Оглавление

[Список сокращений 3](#_Toc11117960)

[Введение 4](#_Toc11117961)

[1 Анализ предметной области 5](#_Toc11117962)

[1.1 Анализ существующих программных решений 6](#_Toc11117963)

[2 Проектирование программной системы 10](#_Toc11117964)

[2.1 Сбор и обработка телеметрических данных 11](#_Toc11117965)

[2.2 Взаимодействие пользователя с системой 14](#_Toc11117966)

[2.3 Прототип 14](#_Toc11117967)

[2.4 Предварительный анализ данных 18](#_Toc11117968)

[3 Разработка 21](#_Toc11117969)

[3.1 Медианный фильтр 21](#_Toc11117970)

[3.2 Разработка базы данных 24](#_Toc11117971)

[3.3 Алгоритмы обнаружения сливов и заправок 26](#_Toc11117972)

[3.3.1 Обнаружение сливов с помощью кубического сплайна. 26](#_Toc11117973)

[3.3.2 Обнаружение сливов с помощью формулы интегрирования 28](#_Toc11117974)

[3.4 Анализ данных в окрестности точек на предмет сливов/заправок 29](#_Toc11117975)

[4 Обработка результатов 31](#_Toc11117976)

[4.1 Юстировка параметров 34](#_Toc11117977)

[4.2 Исследование скорости изменения топлива 35](#_Toc11117978)

[4.3 Тестирование методом черного ящика (Test-case) 36](#_Toc11117979)

[4.4 Диаграмма вариантов использования (Use-case) 38](#_Toc11117980)

[Заключение 40](#_Toc11117981)

[Список источников 41](#_Toc11117982)

# Список сокращений

ТС – транспортное средство.

БД – база данных.

ДУТ – датчик уровня топлива.

ТБ – телекоммуникационные блоки.

ТРК – топливораздаточная колонка.

DFD – Data Flow Diagrams.

# Введение

Современное управление автомобильным парком компаний-перевозчиков основывается на мониторинге автомобилей, по средствам сетей сотовой связи и технологиях *GPS*/*GLONASS*. Результат такого мониторинга позволяет следить за передвижением автомобилей, оценивать ситуация на дорогах, оповещать об авариях, следить за нарушением правил дорожного движения, строить отчеты о необходимых аспектах передвижения транспорта.

Данная работа посвящена совершенствованию такого важного инструмента мониторинга, как построение отчета о расходе топлива. Стоимость топлива – важная составляющая расходов в сферах транспорта, строительства, сельском хозяйстве и пр. Отслеживание уровня топлива позволяет выявлять увеличенный расход одних транспортных средств по сравнению с другими, неисправности транспортных средств, нарушения на заправочных станциях, несанкционированные сливы топлива и пр. Устранение подобных потерь приводит к росту эффективности компании.

# 1 Анализ предметной области

Дизельное топливо – это продукт перегонки нефтяного сырья в виде углеводородов с высокой температурой кипения. Фракционный состав такого горючего определяет его основные характеристики, которые влияют на эффективность работы дизеля.

Расход топлива – уменьшение уровня топлива в течение некоторого промежутка времени. Допустим, де-юре[[1]](#footnote-1) расход топлива машины на 100 км составляет 50 литров. Тогда, де-факто[[2]](#footnote-2) расход топлива на 100 км будет считаться равномерным, если не будет сильно превышать де-юре расход.

Заправка топлива – повышение уровня топлива в баке в течение непродолжительного промежутка времени на минимальный объем заправки (20 – 30 литров).

Слив топлива – уменьшение уровня топлива в течение непродолжительного промежутка времени на минимальный объем слива (5 литров). Когда программа находит потенциальный слив топлива, начинается его диагностика. Если, слив топлива случился в момент стоянки (скорость ТС равна 0), то такое уменьшение топлива считается сливом независимо от его продолжительности, в другом случае данные проверяются на предмет ложных выбросов.

Одной из существенных проблем при определении объема топлива, находящегося в топливных баках, является температурное расширение. Все вещества имеют свойство расширятся при увеличении температуры. Это свойство описывается линейной функцией. Данное физическое свойство вносит погрешности в измерениях, что приводит к ошибкам в результатах измерения. Для точного определения объема топлива в баке, необходимо знать температурный коэффициент расширения и значение температуры. Исходя из этих параметров необходимо учитывать погрешность, получаемую с датчиков уровня топлива, и вносить соответствующие корректировки.

Коэффициент расширения топлива (бензина) – 0,00124 на каждый градус. Все проверки заправочных колонок производятся при температуре +: т.е. ТРК должна налить в мерный бидончик такое количество жидкости, которое при нагревании до + будет ровно по метке. И если топливо холодное, то оно сжимается, становится более плотным, поэтому зимой получается меньше литров, чем летом. Например, 1325 литров летом при +20 будет равно 1240 литров зимой при -20.

## 1.1 Анализ существующих программных решений

В настоящее время редко можно встретить бесплатные решения данной проблемы, в связи с чем было принято реализовать некоторые алгоритмы обработки информации с ДУТ и сделать их общедоступными. Компании не предоставляют алгоритмы обработки своих устройств в открытый доступ, поэтому был произведен обзор только их продукции и ее стоимости.

**АвтоГраф**

АвтоГраф – это система контроля расхода топлива и мониторинга ТС. Предлагает большой спектр услуг от установки датчиков топлива и обработки полученных данных до защиты от угона ТС. Стоимость за одну единицу оборудования 11 500 рублей + 500 рублей ежемесячное обслуживание (с одной ТС). На рисунке 1 изображены основные функции компании.



Рисунок 1 Компания АвтоГраф. Основные функции

**Omnicomm**

Omnicomm – официальный дилер российского производителя систем ГЛОНАСС/GPS мониторинга транспорта и контроля расхода топлива.

Дилерская сеть Omnicomm охватывает все федеральные округа России и СНГ и предлагает большой выбор датчиков уровня топлива. Стоимость за одну единицу оборудования 13 800 рублей + установка датчика 3500 рублей (с одной ТС). На рисунке 2 изображен датчик уровня топлива Omnicom LLS 4.



Рисунок 2 Датчик уровня топлива Omnicomm LLS 4

**Технотон**

Технотон – производит и внедряет собственные решения контроля расхода топлива и мониторинга транспорта: расходомеры топлива DFM, датчики уровня топлива DUT-E, бесконтактные считыватели Crocodile, интерфейсы данных автомобиля MasterCAN (см. рисунок 3). Стоимость за одну единицу оборудования 7000 рублей + 2000 рублей установка датчика (с одной ТС).



Рисунок 3 Устройства для контроля расхода топлива компании Технотон

**Gurtam**

Gurtam ‒ занимает около 36% рынка мониторинга коммерческого транспорта в странах СНГ и активно продвигаются на рынок Европы, Ближнего Востока, США, Латинской Америки, Африки, Австралии и даже Новой Зеландии. Занимается разработкой программного обеспечения для спутникового GPS/ГЛОНАСС мониторинга и управления транспортом – Wialon (см. рисунок 4).

Затраты на Wialon Pro составят:

* Лицензии: 2300 евро (Wialon Pro в базе) + 1000 евро (модуль «Отчеты») + 500 евро (модуль «Уведомления») + 400 евро (модуль «Задания») + 500 евро (модуль «Расширенные отчеты») + 900 евро (модуль ActiveX) + 8 \* (пакет расширения на 25 объектов) \* 500 евро = 9600 евро;
* Обновление Wialon Pro: 30% от прайсовой стоимости приобретенных лицензий: 2880 евро;
* Техническая поддержка на уровне Unlimited в течение второго года: 1500 евро;
* Аренда сервера в датацентре: 200 евро \* 24 месяца = 4800 евро.

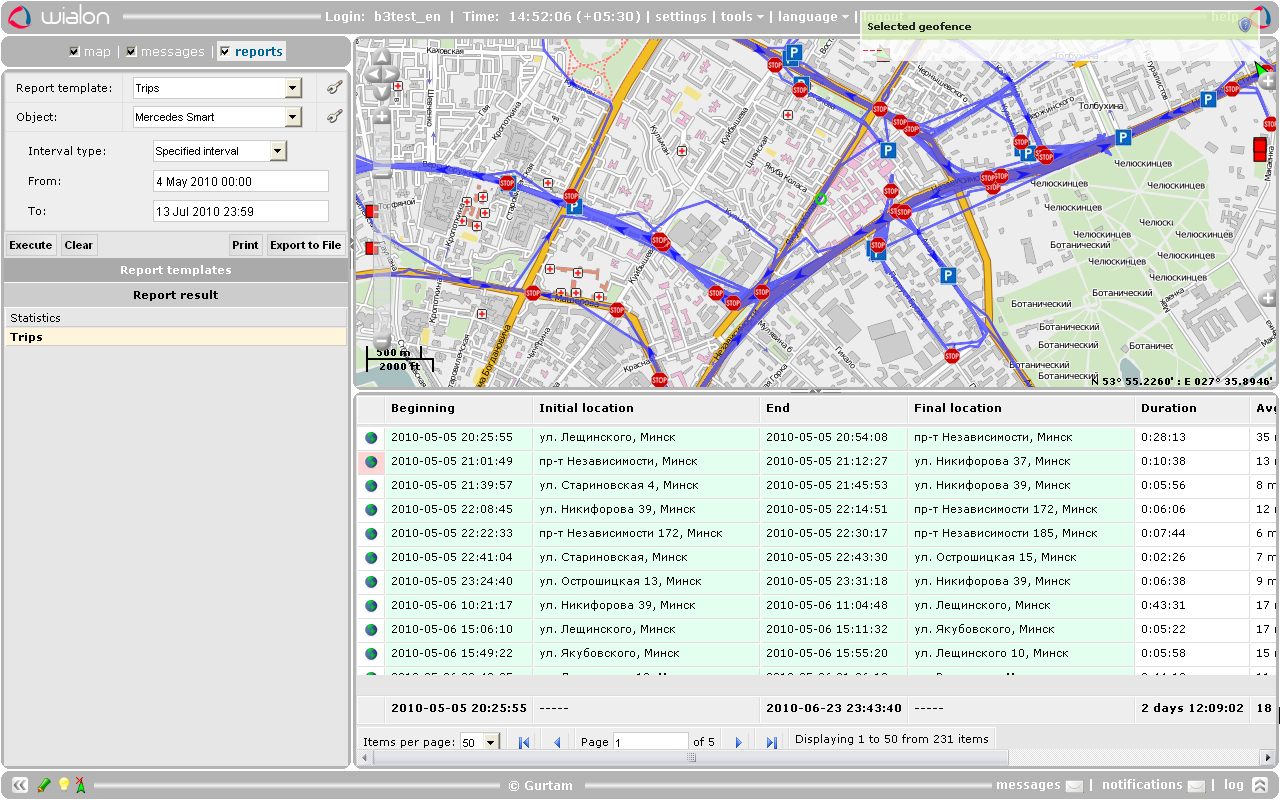


Рисунок 4 Скриншот программы для спутникового мониторинга транспорта Wialon

# 2 Проектирование программной системы

Мониторинг расхода топлива и последующий анализ позволяет выявлять факт недобросовестного его использования и хищения. Разработка программной системы по выявлению недобросовестного использования топлива является целью данной работы.

**Требования к разрабатываемым алгоритмам**

Предлагаемые алгоритмы ориентированы на решение следующих задач:

* выявление сливов и заправок топлива по исходным данным;
* построение отчета по топливу за запрашиваемый промежуток времени.

**Системные требования**

Необходимо иметь среду разработки Python.

**Технические требования**

Для функционирования системы необходимо техническое обеспечение со следующими минимальными характеристиками:

* оперативная память 256 Мб или больше;
* свободное место на жёстком диске от 50 Мб;
* архитектура с разрядностью 32 бит или 64 бит (x86 или x64);
* операционная система Windows XP, Windows Vista, Windows 7, Windows 8.

**Отчет должен содержать:**

* Время начала и окончания отчета;
* Уровень топлива в баке к началу периода отчета и по окончании периода отчета;
* Список событий заправок и сливов с указанием:
* Времени и длительности события;
* Вида события (слив/заправка);
* Изменения объема топлива в литрах в результате события;
* Начальный уровень топлива – объём (уровень) топлива в баке непосредственно перед событием;
* Конечный уровень топлива – объём (уровень) топлива в баке непосредственно после события.
* Полный объем расхода топлива за период события (с учетом сливов и заправок).

**Технические требования к отчету:**

* Точность определения объемов событий не должна быть хуже 10 л;
* Точность определения времени события не должна быть хуже 1 мин;
* Отчет не должен содержать пропусков событий объемом более 20 л;
* Отчет не должен содержать ложных событий.

Ориентированность на нишу бесплатного и общедоступного средства анализа данных по расходу топлива является главенствующим фактором при выборе средств реализации и развертывания.

Программная система базируется на концепции клиент-серверного приложения с тонким клиентом, означающая что от устройств клиента требуется только подключение к сети интернет и возможность визуализация передаваемых сервером данных, требовательные по производительности расчеты проводятся на стороне сервера.

## 2.1 Сбор и обработка телеметрических данных

На ТС установлены ТБ. Каждый блок умеет:

* устанавливать интернет соединения с сервером (поверх GSM по IP адресу) и передавать данные по протоколу TCP/IP;
* определять текущие координаты и скорость (GPS/ГЛОНАСС); (GPS/ГЛОНАСС данные могут быть и недоступны – закрытые помещения, тоннели, городская инфраструктура, защищенные зоны);
* к блоку могут подключаться внешние устройства. Датчики формируют сигнал в mV. ТБ передает эти значения и код своего входа на сервер;
* ТБ может кешировать данные в локальной постоянной памяти. Поэтому, даже если нет GSM данные не пропадают.

Отключение блока от питания – быстро разряжает внутренний аккумулятор и приводит к неработоспособности блока – это выражается в виде отсутствия данных. Часто на грузовых машинах снимают клемму с аккумулятора на некоторое количество времени, во избежание разрядки. После того, как клемму возвращают на место, блок начинает работать в течение 10 минут. Обычно, машина за это время успевает проехать 3-5 км. Иногда сигнал отсутствует и по другим причинам, начиная от нарушения контактов и оканчивая поломкой ТБ.

После того, как данные попадают на сервер, они ассоциируются с конкретным ТС. В данном случае, это данные GPS/ГЛОНАСС и данные от датчика топлива.

Для каждого ТС с датчиком топлива имеется тарировочная таблица. Данные для нее были получены таким образом:

* ТС устанавливалась на горизонтальную площадку;
* топливо заливалось мерной емкостью и записывались показания датчика (соответствующего выхода).

Таблицу помещают на сервер и связывают с конкретным ТС и датчиком. Таблица позволяет переводить данные из mV в литры для конкретного ТС.

Данные об уровне топлива, датчиках и времени в течение, которого машина отслеживалась поступают в виде файла Excel и устроены так:

1. Номера машин, для которых извлекались данные по топливу за последние несколько месяцев;
2. Дальше идут группы по 5 колонок. Каждая группа содержит данные от одной машины;
3. Номер машины;
4. Unix-timestamp (количество секунд с 1 янв. 1970 г) времени, на которое был измерен уровень топлива;
5. Unix-timestamp времени, на которое данные были записаны в БД;
6. Код датчика, который прислал данные;
7. Значение топлива в mV, которое прислал датчик.
8. Данные получены от 37 машин. Объем данных составлял более 9 млн записей. Но тарировочные таблицы имеются только для 32 машин, поэтому после загрузки данных в БД, записи машин для которых отсутствовали тарировочные данные, были исключены из рассмотрения. В итоге рассматривается массив данных размером более 5 млн данных для 32 машин;
9. Кроме того, для некоторых машин есть 1 датчик, для других 2, но поскольку тарировочные таблицы получены только для одного датчика – данные от второго исключены из рассмотрения;

* Датчики с кодом 65-66 генерируют значения от 0 до 4095. 0 – 0 литров, 4095 – 1000 л;
* Датчики с кодом 16-17 генерируют значения от 0 до 2000. 0 – 0 литров, 2000 – 1000 л.

В отдельных архивах данные по скорости движения каждой машины и тарировочные таблицы для перевода расхода топлива из mV в литры (для каждой машины таблица уникальна).

Первичные данные разбиты вручную по отдельным Excel файлам для каждой машины.

## 2.2 Взаимодействие пользователя с системой

Для доступа к системе необходимо зайти на сайт системы и пройти процедуру регистрации. Для прохождения требуется заполнить электронную форму, в которой, в частности, требуется ввести email и пароль, по которым и производится авторизация в системе.

Авторизовавшись в системе, пользователь перенаправляется на главную страницу, отображающую архивные данных его прошлых запросов и возможность создать новый запрос.

## 2.3 Прототип

**Вход**

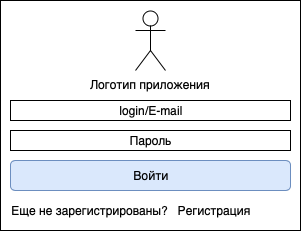


Рисунок 5 Окно входа

Описание:

В данном окне осуществляется вход в приложение с помощью ввода логина и пароля, для входа предусмотрена кнопка «Войти». Если пользователь еще не зарегистрирован в приложении, то у него есть возможность перейти в окно регистрации и пройти данную процедуру.

**Регистрация**

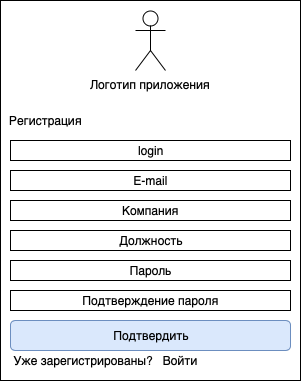


Рисунок 6 Окно регистрации

Описание:

В данном окне осуществляется регистрация пользователя в приложении. Подразумевается ввод: логина, e-mail, компании, занимаемой должности, пароля, а также подтверждение пароля.

Предусмотрена кнопка «Подтвердить», а также кнопка «Войти», если пользователь ранее уже был зарегистрирован.

**Личный кабинет**



Рисунок 7 Личный кабинет

Описание:

В личном кабинете предоставляется некоторая информация о пользователе. Также предусмотрен следующий функционал: создание нового отчета, просмотр сохраненных отчетов, редактирование данных пользователя.

**Создание нового отчета**

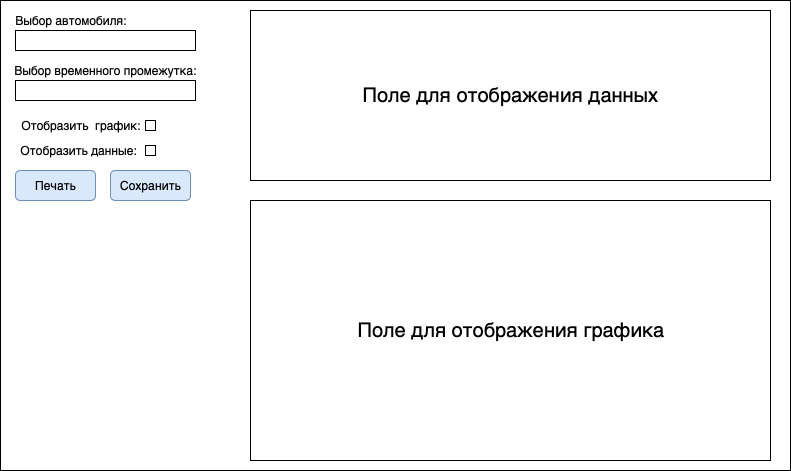


Рисунок 8 Создание нового отчета

Описание:

В данном окне можно осуществлять выбор автомобиля, а также промежуток времени, за который требуется просмотреть отчет. Также есть возможность отобразить график или вывести текстовое отображение данных. Присутствуют кнопки с помощью которых можно распечатать отчет или же сохранить его.

**Окно сохраненных отчетов**



Рисунок 9 Сохраненные отчеты

Описание:

В данном окне отображается список сохраненных отчетов.

**Редактирование данных пользователя**

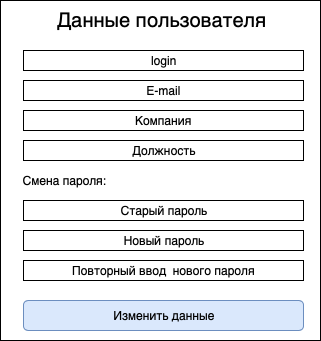


Рисунок 10 Окно редактирования данных пользователя

Описание:

В данном окне предусмотрено изменение данных пользователя: логина, e-mail, компании, должности, пароля.

## 2.4 Предварительный анализ данных

Прежде чем приступать к разработке алгоритма, необходимо выявить проблемы, которые возникают в реальных ситуациях и мешают построению отчета. С этой целью был проведен предварительный анализ данных.

В результате такого анализы были выявлены следующие проблемы:

* Датчики уровня топлива время от времени выдают случайные выбросы уровня;
* Уровень топлива меняется в зависимости от температуры;
* Топливная система имеет некоторую инерцию – сигнал об уровне топлива при значительных изменениях (заправках) имеет время релаксации;
* Показания датчиков зависят от наклона бака;
* Случаются периоды значительного отсутствия телеметрических данных. Периодически отсутствует сигнал *GPS*.

Ниже приводятся соответствующие примеры.

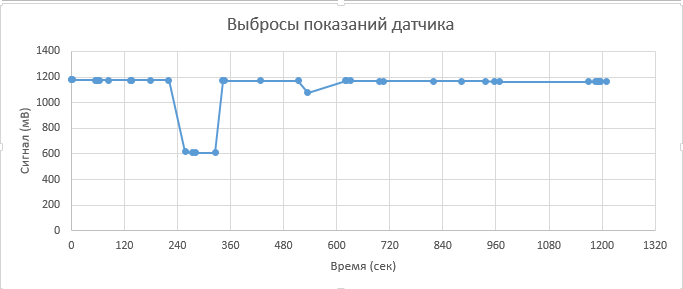


Рисунок 11 Пример выброса показаний датчика

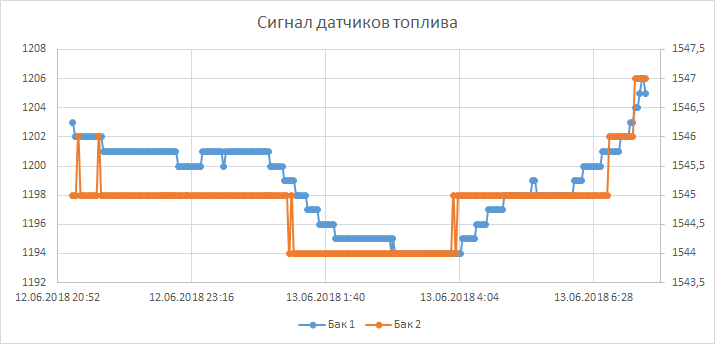


Рисунок 12 Пример изменения уровня топлива в баках от температуры. Баки остывали ночью и нагревались к утру, когда автомобиль стоял.



Рисунок 13 Медленный выход показаний на окончательный уровень топлива в баке.

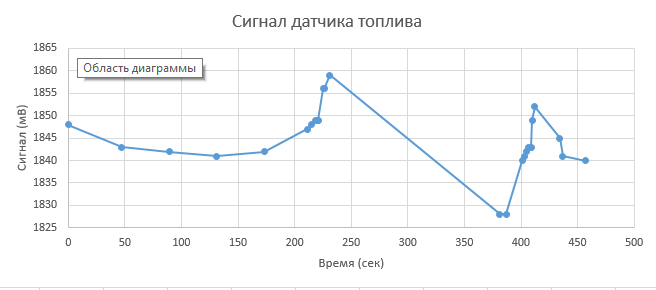


Рисунок 14 Колебания показаний датчика топлива в результате движения топлива в баке.

Очевидно, выявленные помехи существенно влияют на формирование отчета. Если выбрать неправильную точку отсчета при определении объема заправки или слива, то ошибка может достигать 50 л. В некоторых случаях, как (Рисунок 14) можно принять снижение уровня топлива в течении 3 мин за слив, если не обратить внимание на его последующее повышение.

Аналогично, если не учесть изменение температуры топлива в процессе заправки, то показания могут отличаться от объема фактически заправленного топлива.

# 3 Разработка

## 3.1 Медианный фильтр

Одной из проблем первичных данных является то, что данные флуктуируют (колеблются). Например, это можно увидеть на следующем рисунке, где на фоне показаний датчика появляется резкий всплеск в одно значение.

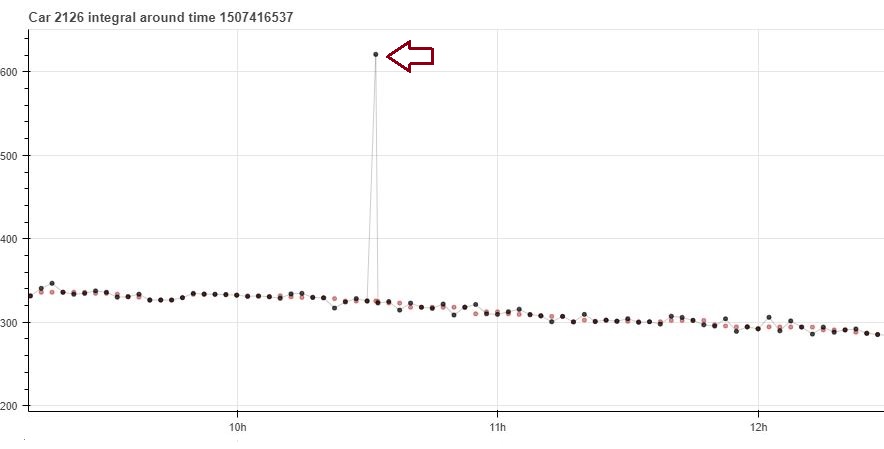


Рисунок 15 Пример флуктуации в первичных данных

Если такие данные не отфильтровать, то впоследствии применяемые методы интерполяции будут давать ложно детектируемые случаи заправки/слива. Для фильтрации таких случаев был использован медианный фильтр.

Особенностью данного фильтра является селективность по отношению к тем элементам массива, которые представляют собой немонотонную последовательности чисел в пределах окна (апертуры) фильтра, и резко выделяющихся на фоне соседних отсчетов. В то же время на монотонную составляющую последовательности медианный фильтр не действует, оставляя её без изменений. Благодаря этой особенности, медианные фильтры при оптимально выбранной апертуре могут сохранять без искажений резкие границы объектов, эффективно заглушая некоррелированные или слабо коррелированные помехи.

Для решения данной задачи был выбран медианный фильтр с окном 5, который позволяет отфильтровывать выбросы в одну-две точки. Алгоритм этого метода состоит в том, что он сортирует данные (в пределах выбранного окна) по возрастанию и выбирает среднее значение отсортированной последовательности (на рисунке 16 приведён код метода обработки медианным фильтром).

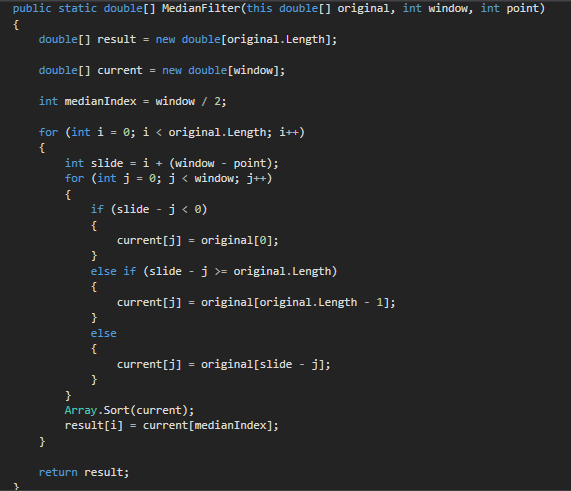


Рисунок 16 Реализация алгоритма медианной фильтрации

Из плюсов метода можно отметить:

* простая структура фильтра для программной реализации;
* не изменяет ступенчатые и пилообразные функции;
* хорошо заглушает одиночные шумовые выбросы.

Из недостатков медианных фильтров можно отметить:

* фильтр вызывает уплощение вершин треугольных функций;
* при увеличении размеров окна фильтра происходит размытие сильных изменений сигнала и скачков.

Пример работы медианного фильтра можно увидеть на рисунках 17 и 18.

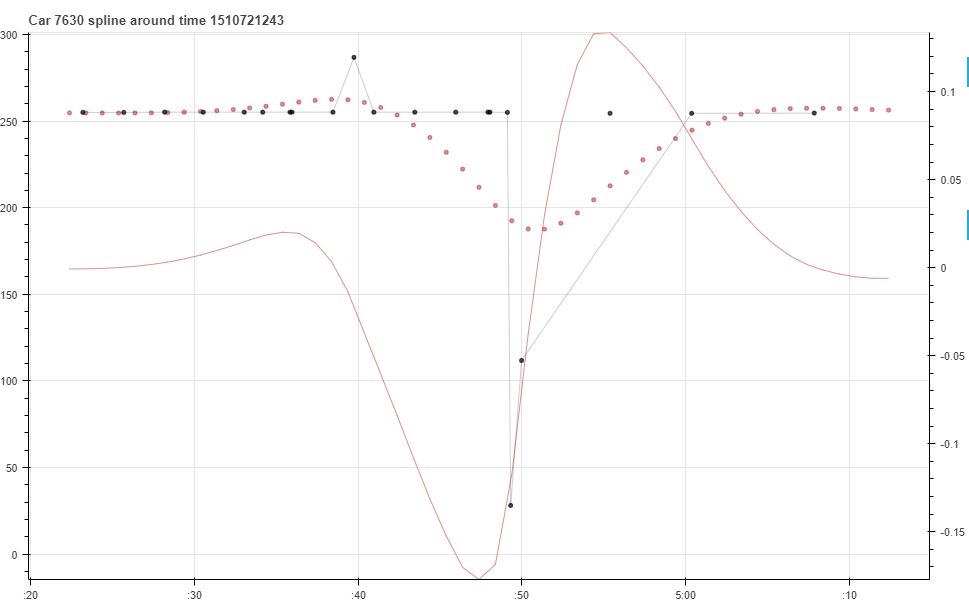


Рисунок 17 График до работы медианного фильтра[[3]](#footnote-3)

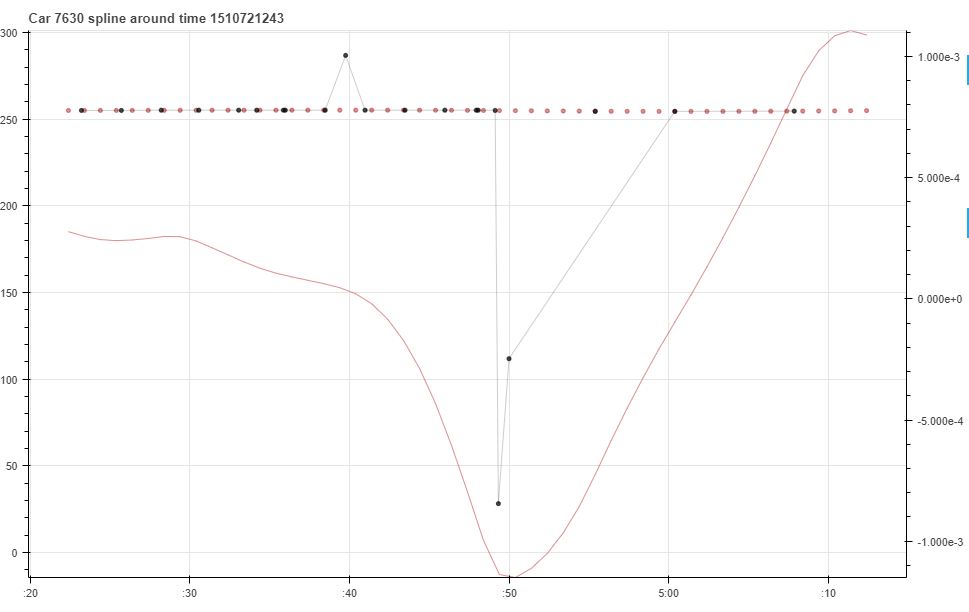


Рисунок 18 График после работы медианного фильтра[[4]](#footnote-4)

## 3.2 Разработка базы данных

Данные загружались в базу данных программой *LoadToDB*, написанной на языке программирования *C*#. В качестве СУБД для исследований была выбрана *SQLite*. Это встраиваемая кроссплатформенная БД, поддерживающая практически весь набор команд *SQL*, что позволяет масштабировать решение на промышленные СУБД типа *Postgres*, *SQL Server* и т. п. поддерживающие язык структурированных запросов *SQL*.

**Схема БД:**

1. *Car (CID INTEGER PRIMARY KEY, DID INTEGER NOT NULL, XMLParam TEXT NOT NULL)*. *CID* – первичный ключ, *DID* – номер машины под которым указаны данные (1547, 116208 и т.п.), *XMLParam* – тарировочные данные;
2. *Fuel (CID INTEGER PRIMARY KEY, DID INTEGER NOT NULL, GPSTime INTEGER NOT NULL, LoadTime INTEGER NOT NULL, Sensor INTEGER NOT NULL, Level REAL NOT NULL, Moving INTEGER, LevelTT REAL, LevelM REAL). CID* – первичный ключ, *DID* – номер машины под которым указаны данные (1547, 116208 и т.п.), *GPSTime* – время *GPS*, *Sensor* – данные датчика, *LoadTime –* время, в которое данные были записаны в БД, *Sensor* – данные датчика, *Level* – уровень топлива в *mV*, *Moving* – скорость машины, *LevelTT* – уровень топлива после тарировки датчиков, *LevelM* – медианный фильтр по данным топлива (см. рисунок 19)*;*
3. *Fuel\_s* – (*DID INTEGER NOT NULL REFERENCES Car (DID), GPSTime INTEGER NOT NULL, Level INTEGER* (0, 4095) *NOT NULL, LevelTT REAL* (0, 4095) *NOT NULL, Moving INTEGER*). *DID* – номер машины под которым указаны данные (1547, 116208 и т.п.), *GPSTime* – время *GPS*, *Level* – уровень топлива в *mV*, *LevelTT* – уровень топлива после тарировки датчиков, *Moving* – скорость машины. Имеет составной первичный ключ по двум полям *DID* и *GPSTime*;
4. *Speed* (*CID*, *DID*, *GPSTime*, *Value*). *CID* – первичный ключ, *DID* – номер машины под которым указаны данные (1547, 116208 и т.п.), *GPSTime* – время *GPS*, *Value* – скорость в милях. Является временной таблицей, после занесения полей в таблицу *Fuel\_s* не используется.

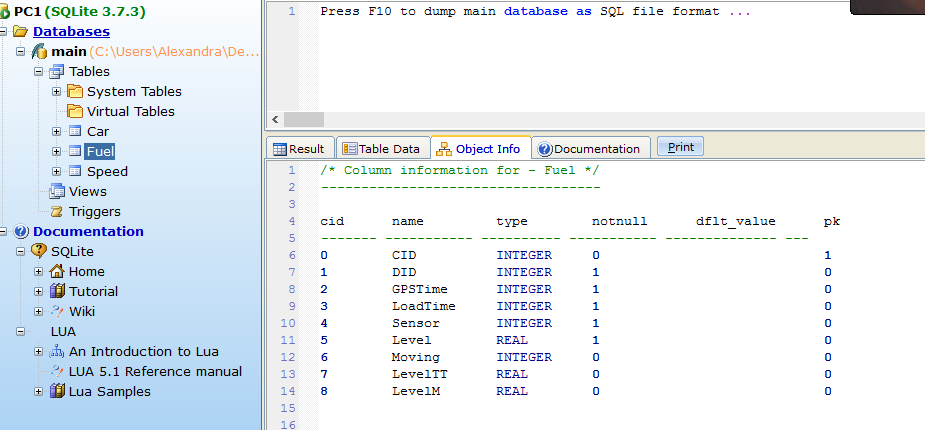


Рисунок 19 Пример БД. Описание полей в таблице Fuel

В таблице *Fuel* находятся все данные, в таблице *Fuel\_s* находятся отфильтрованные данные о машинах, которые имеют тарировочные таблицы.

Загруженные данные разбивались программой *SplitOnSegments*,написанной на языке программирования *C*# на сегменты непрерывных промежутков данных в разрезе каждой машины. Эта информация записывалась в таблицу *sgmts*\_1000:

*CID INTEGER PRIMARY KEY, DID INTEGER NOT NULL, Start INTEGER NOT NULL, End INTEGER NOT NULL*. *CID* – первичный ключ, *DID* – номер машины (1547, 116208 и т.п.), *Start* – начало промежутка, *End* – конец промежутка непрерывных данных.

## **3.3 Алгоритмы обнаружения сливов и заправок**

Были рассмотрены несколько алгоритмов обнаружения сливов и заправок ТС. Далее будет рассмотрен каждый из них.

### 3.3.1 Обнаружение сливов с помощью кубического сплайна.

Некоторая функция *fx* задана на отрезке [*a*, *b*], разбитом на части . Кубическим сплайном дефекта 1 (разность между степенью сплайна и получившейся [гладкостью](https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D0%B0%D0%B4%D0%BA%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F)) называется функция , которая:

* на каждом отрезке является многочленом степени не выше третьей;
* имеет непрерывные первую и вторую производные на всем отрезке ;
* в точках выполняется равенство , т.е. сплайн интерполирует функцию *f* в точках .

С помощью программы *SplineToDb* первоначальные данные интерполируются сплайнами в разрезе каждого сегмента с шагом в одну минуту и сохраняются в БД в таблице *spline*\_1 со структурой:

*CID INTEGER PRIMARY KEY, DID INTEGER NOT NULL, GPSTime INTEGER NOT NULL, Value REAL NOT NULL, Der1 REAL NOT NULL, Der2 REAL NOT NULL*, где *CID* – первичный ключ, *DID* – номер машины (1547, 116208 и т.п.), *GPSTime* – время *GPS, Value –* значение сплайна в данной временной точке, *Der1* – первая производная, *Der2* – вторая производная.

Статистическая обработка полученных данных осуществляется с помощью библиотек *NumPy* и *Pandas* на языке *Python*:

* из таблицы *spline*\_1 загружаются данные в разрезе машины и сегмента (см. приложение 4 функцию *load\_data\_by\_segment*);
* находятся локальные минимумы первой производной и отсекаются по эмпирически найденному пороговому значению;

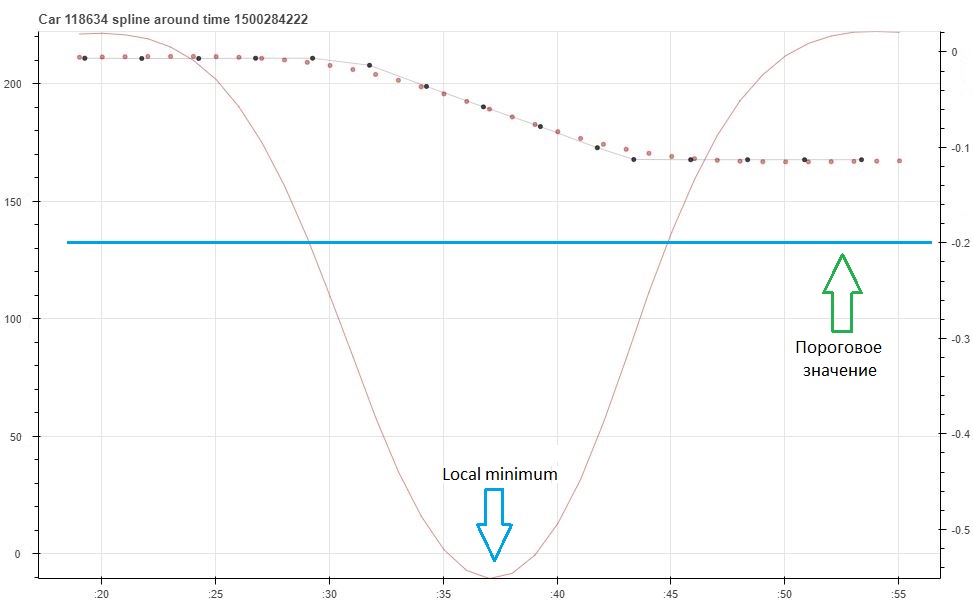


Рисунок 20 Локальный минимум первой производной в точке[[5]](#footnote-5)

* В окрестностях найденных локальных минимумов проводится исследование исходных значений на предмет слива/заправки топлива.

Особенностью интерполирования с помощью сплайнов является то, что при резком изменении (скачке данных) интерполированные сплайном значения могут «нырять» в противоположную сторону перед/после изменения. Эта особенность приводит к тому, что при выбранном пороговом значении заправки также попадают в список локальных минимумов (см. рисунок 21).

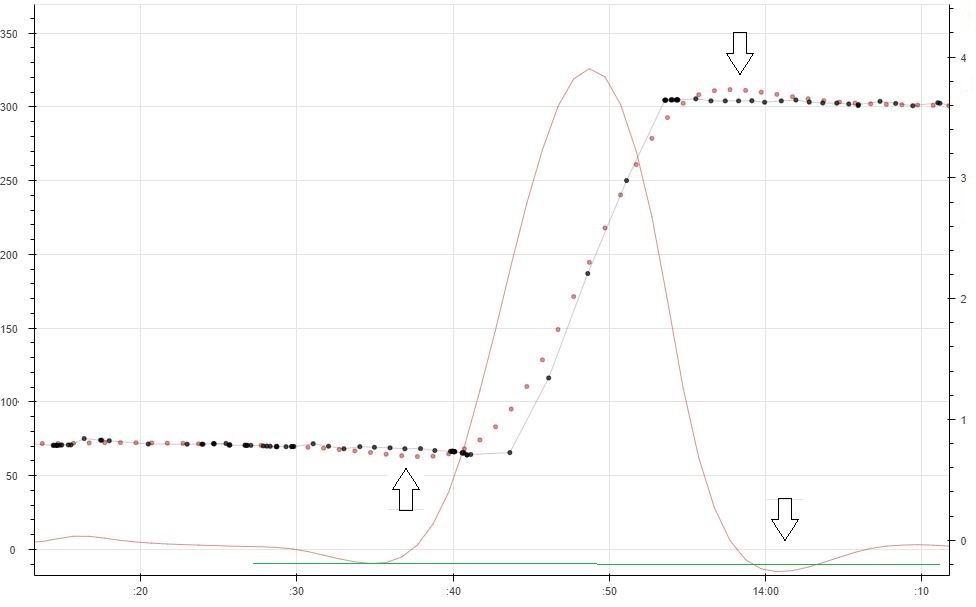


Рисунок 21 Поведение сплайна при скачке данных[[6]](#footnote-6)

### 3.3.2 Обнаружение сливов с помощью формулы интегрирования

Данный метод основан на алгоритме, предложенном в статье. Алгоритм состоит в следующем:

* данные в разрезе каждой машины и сегмента загружаются для обработки; применяется медианный фильтр с окном 5 для удаления всех выбросов шириной в одну-две точки;
* полученная последовательность данных интерполируется с шагов в 30 секунд для упрощения дальнейших операций;
* рассчитывается разница по границам окна по формуле

|  |  |
| --- | --- |
| , | (1) |

где – значение уровня топлива (после медианного фильтра) в *n*-й точке, – полученное значение расхода топлива, *w* – половина ширины выбранного окна. Ширина окна выбирается эмпирически;

* проводится суммирование в соответствии с формулой (1) по той же ширине окна, которая была выбрана на предыдущем этапе.

|  |  |
| --- | --- |
| , | (2) |

где – значение расхода топлива, – значение, полученное на предыдущем шаге, *w* – половина ширины выбранного окна.

* Находятся локальные минимумы значений, полученных по формуле (2) и отсекаются по эмпирически найденному пороговому значению;
* В окрестностях найденных локальных минимумов проводится исследование исходных значений на предмет слива/заправки топлива.

## 3.4 Анализ данных в окрестности точек на предмет сливов/заправок

В результате применения перечисленных выше методов получается список возможных заправок/сливов. Для нахождения точных значений начала и конца заправки/слива и их объема топлива в литрах применяется метод исследования исходных данных в окрестностях указанных точек. Исследование проводится с помощью статистических пакетов *NumPy* и *Pandas* на языке программирования *Python*.

Сперва детектируются ложные срабатывания датчиков, которые не были отфильтрованы медианным фильтром (см. рисунок 22).

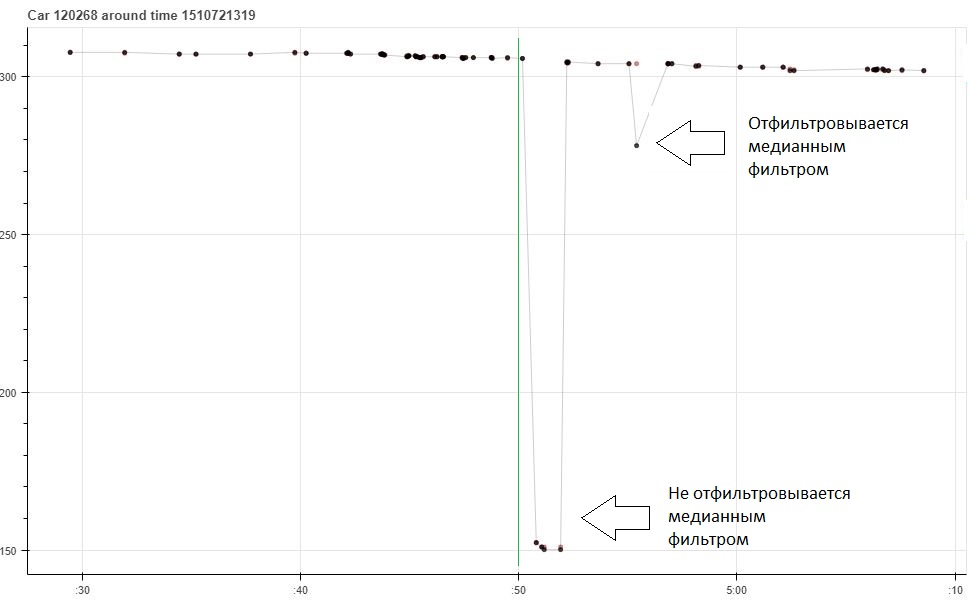


Рисунок 22 Пример ошибочного попадания в список сливов/заправок[[7]](#footnote-7)

Для этого строятся -окрестности исследуемой точки и среди них находятся такие-окрестности, в которых разброс значений будет сопоставим с разницей между максимальным и минимальным значением на всем рассматриваемом промежутке. Если среди этих окрестностей находится окрестность, для которой разница между средними значениями слева и справа меньше порогового значения, выбираемого эмпирически, то эта точка рассматривается как ложное срабатывание и не попадает в список заправок/сливов.

# 4 Обработка результатов

Весь набор данных по топливу был обработан интегральным методом и методом сплайнов в двух вариантах - с предварительной обработкой данных медианным фильтром и без обработки. Пороговые значения для поиска локальных экстремумов целевых функций этих методов подбирались эмпирически.

Ложными срабатываниями считались такие локальные экстремумы целевой функции, при которых разница средних значений левой и правой границ окрестности этого экстремума не превышала определенной границы.

Повышение уровня топлива или заправка - локальный экстремум целевой функции в окрестности которого уровень топлива в баке повысился.

Уменьшение уровня топлива или слив - локальный экстремум целевой функции в окрестности которого уровень топлива снизился.

Первая таблица показывает результаты обработки данных при поиске по локальным минимумам.

Таблица 1. Результаты обработки данных. Поиск по локальным минимумам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Ложные срабатывания  *df*[*df.Diff*==0] | Повышения уровня топлива  *df*[*df.Diff*>0] | Уменьшения уровня топлива  *df*[*df.Diff*<0] | Всего локальных минимумов  *len*(*df*) |
| Интегральный с медианным фильтром  (порог -20) | 66 | 2 | 177 | 245 |
| Интегральный без медианного фильтра  (порог -20) | 126 | 5 | 215 | 346 |
| Сплайн с медианным фильтром  (порог -0.05) | 474 | 18 | 70 | 562 |
| Сплайн без медианного фильтра  (порог -0.05) | 388 | 33 | 94 | 515 |

Как видно из таблицы интегральный метод без предварительной обработки медианным фильтром дает большое количество ложных срабатываний: 126 из 346.

Обработка входных данных медианным фильтром позволяет в значительно уменьшить количество рассматриваемых точек и, соответственно, ложных срабатываний: 245 против 346 до обработки медианным фильтром.

Оба метода могут только косвенно указывать на заправки (повышение уровня топлива) при поиске локальных минимумов целевых функций. Поэтому данные, попавшие в столбец заправок, могут представлять из себя как выбросы данных, так и заправки, попавшие в окрестность локального минимума.

Вторая таблица показывает результаты обработки данных при поиске по локальным максимумам.

Таблица 2. Результаты обработки данных. Поиск по локальным максимумам

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Ложные срабатывания  *df*[*df*.*Diff*==0] | Повышения уровня топлива  *df*[*df.Diff*>0] | Уменьшения уровня топлива  *df*[*df.Diff*<0] | Всего локальных максимумов  *len*(*df*) |
| Интегральный с медианным фильтром  (порог 25) | 11 | 1534 | 3 | 1548 |
| Интегральный без медианного фильтра  (порог 25) | 30 | 1560 | 4 | 1594 |
| Сплайн с медианным фильтром  (порог 0.15) | 10 | 1407 | 1 | 1418 |
| Сплайн без медианного фильтра  (порог 0.15) | 15 | 1419 | 7 | 1441 |

## 4.1 Юстировка параметров

Большое влияние на получаемые результаты оказывает настройка параметров каждого из рассматриваемых методов.

Для сплайн-метода параметрами настройки были параметр сглаживания *rho* и пороговые значения для локальных максимумов и минимумов *threshold* (*spline\_min\_threshold* и *spline\_max\_threshold*). При нулевом значении *rho* сплайн проходит по точкам данных. При больших значениях параметра сглаживания *rho* получается максимально гладкая кривая, которая минимизирует отклонения от данных. Пороговые значения задают чувствительность метода - чем они меньше, тем меньшие изменения данных попадут в рассмотрение.

Интегральный метод настраивается пороговыми значениями для локальных экстремумов (*integral\_min\_threshold* и *integral\_max\_threshold*) и размером окна интегрирования (integral\_window\_size).

Например, для подбора параметра порогового значения *integral\_max\_threshold* были взяты эталонные данные нескольких машин. Был осуществлен прогон по параметру *integral\_max\_threshold* в диапазоне от 1 до 300. Выяснилось, что самым оптимальным пороговым значением является 25. При таком задании параметра программа выводила все достоверные увеличения топлива, при этом отбрасывая все ложные срабатывания.

Аналогичным образом находился параметр *integral\_min\_threshold.*

## 4.2 Исследование скорости изменения топлива

При анализе локальных экстремумов находятся временные точки начала и конца расхода/заправки и его объем. Это позволяет вычислить скорость изменения топлива:

**>** *format\_df(df)*

**>** *df.sort\_values(by='FuelS')*

*Out*[42]:

*DID GPSTime        Diff TimeDiff       FuelS*

157  112644  1504259967  -97.117117       42 **-138.738739**

185  116208  1488749963 -225.382164       140 **-96.592356**

183  116208  1488729702 -209.731933       138 **-91.187797**

154  112644  1503925786 -192.432432       150 **-76.972973**

155  112644  1503930608 -190.630631       150 **-76.252252**

176  116208  1487885452  -14.942529      12 -74.712644

153  112644  1503757463 -217.647059       182 -71.751778

168  114743  1504579011 -154.489331       161 -57.573664

…

При сортировке экстремумов по скорости изменения топлива можно выявить физически невозможные скорости слива топлива. Как видно из рисунка 23 это неотфильтрованные медианными фильтром аномалии в данных.

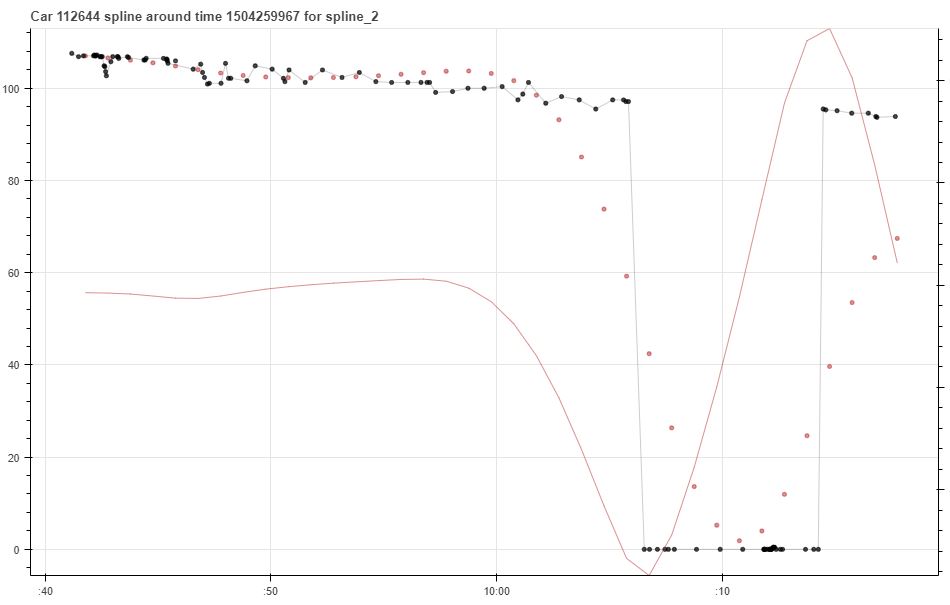


Рисунок 23 Пример аномалии в данных[[8]](#footnote-8)

## 4.3 Тестирование методом черного ящика (Test-case)

Тест-кейс №1: Авторизация

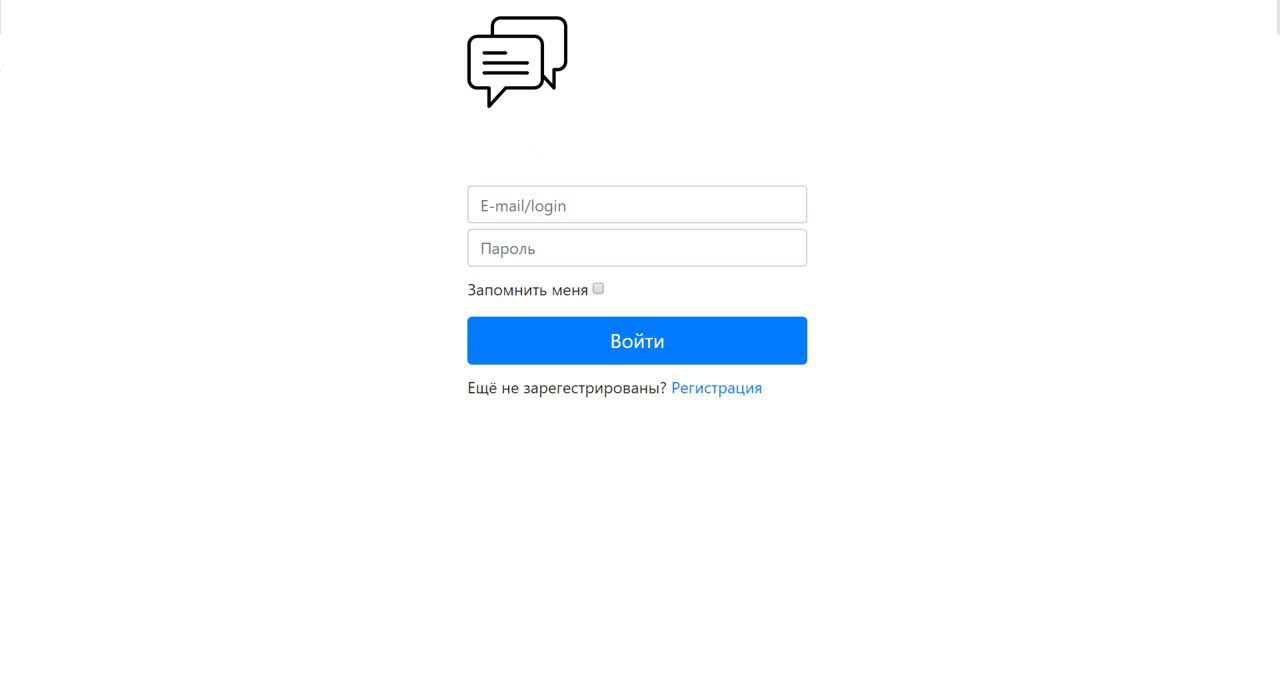


Рисунок 24 Окно авторизации

Шаги:

1. Ввести логин.
2. Ввести пароль.
3. Нажать на кнопку «Войти»

Ожидаемый результат: Успешная авторизация.

|  |  |
| --- | --- |
| Действие | Ожидаемый результат |
| Ввод логина: admin  Ввод пароля: 123456  Нажатие кнопки войти. | Успешная авторизация. |
| Ввод логина: admin  Ввод пароля: 654321  Нажатие кнопки войти. | Ошибка: «Неправильно введен логин или пароль» |

Тест-кейс №2: Регистрация

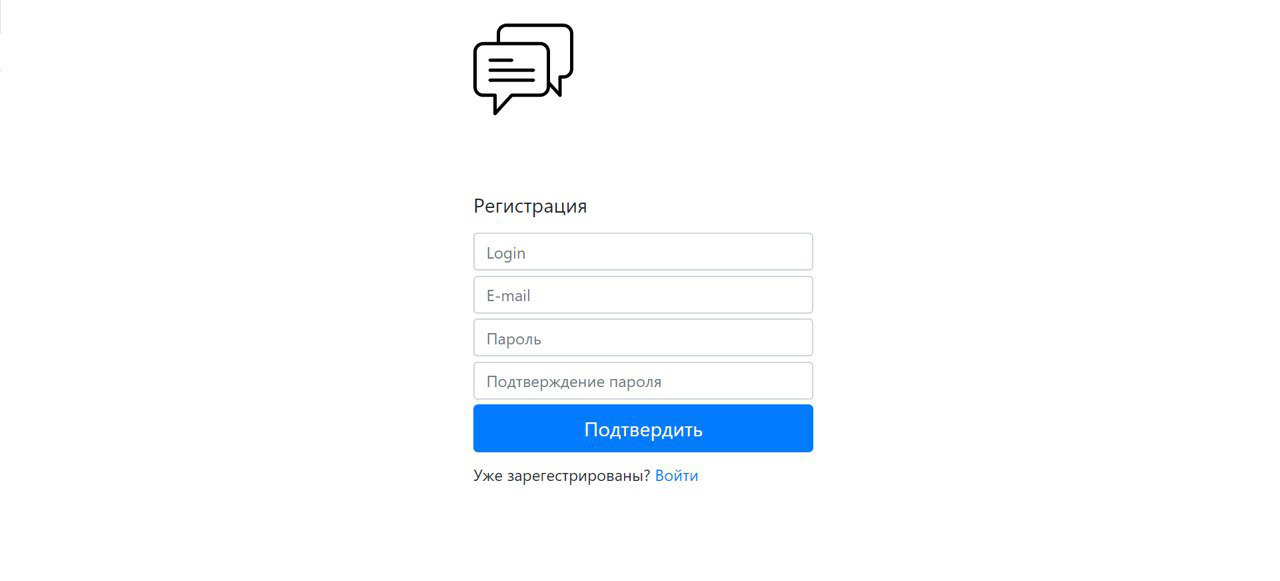


Рисунок 25 Окно регистрации

Шаги:

1. Нажать на кнопку «Регистрация»
2. Ввести логин
3. Ввести E-MAIL
4. Ввести пароль
5. Повторно ввести пароль
6. Нажать кнопку подтвердить

Ожидаемый результат: Успешная регистрация.

|  |  |
| --- | --- |
| Регистрация | Ожидаемый результат |
| Нажать на кнопку «Регистрация»  Ввести логин: admin123  Ввести E-MAIL: admin123@mail.ru  Ввести пароль: 123456  Повторно ввести пароль: 123456  Нажать кнопку подтвердить | Успешная регистрация нового пользователя |
| Нажать на кнопку «Регистрация»  Ввести логин: admin  Ввести E-MAIL: admin@mail.ru  Ввести пароль: 123456  Повторно ввести пароль: 123456  Нажать кнопку подтвердить | Ошибка «Пользователь с таким логином уже существует» |

## 4.4 Диаграмма вариантов использования (Use-case)

Пользователю предлагается пройти авторизацию или регистрацию в приложении. Далее пользователю необходимо заполнить ряд необходимых полей. Диаграмма вариантов использования представлена на рисунке ниже.

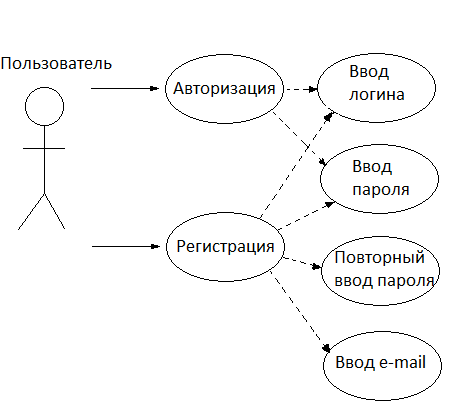


Рисунок 26 Диаграмма вариантов использования

# Заключение

Исследование предметной области и реализация программного средства позволило расширить инструментарий мониторинга расхода топлива и потенциально сократить издержки транспортных компаний.

Программная система позволяет выявлять случаи недобросовестного использования топлива, накапливать и анализировать различные временные промежутки. В процессе работы представлены результаты сравнения нескольких методов анализа данных и их предварительной обработки. Результатом работы программной системы является отчет по расходу топлива за заданный промежуток времени, соответствующий требованиям, указанным при постановке задачи.

# Список источников

1. Шилдт, Герберт. Ш57 C# 4.0: полное руководство.: Пер. с англ. — М.: ООО "И.Д. Вильямс", 2011. — 1056 с.: ил. — Парал. тит. англ.
2. *Python|*статья — *URL*: *http://www.machinelearning.ru/wiki/index.php?title=Python* (дата обращения 23.02.2018).
3. СайтСитиСтрой*Oil|*База знаний *— URL:* [*http://dostavka-toplivo-spb.ru/poleznye-stati/147-vyazkost-dizelnogo-topliva-vidy-i-opredelenie*](http://dostavka-toplivo-spb.ru/poleznye-stati/147-vyazkost-dizelnogo-topliva-vidy-i-opredelenie)(дата обращения 23.04.2019).
4. Электронная библиотека БГУ| Факультет прикладной математики и информатики| Материалы конференций факультета прикладной математики и информатики| Секция 9. Компьютерные технологии в приборостроении — *URL:* [*http://elib.bsu.by/handle/123456789/160530*](http://elib.bsu.by/handle/123456789/160530)(дата обращения 15.04.2019).
5. Национальная библиотека им. Н. Э. Баумана *Bauman National Library|* Медианная фильтрация — *URL:* [*https://ru.bmstu.wiki/Медианная\_фильтрация*](https://ru.bmstu.wiki/Медианная_фильтрация)(дата обращения 04.03.2019).
6. *Python Practice Book* [*Anand Chitipothu*](http://anandology.com/) *— URL:* [*https://anandology.com/python-practice-book/index.html*](https://anandology.com/python-practice-book/index.html)(дата обращения 04.04.2019).
7. Сайт Технотон| Продукты| Датчики уровня топлива *— URL:* [*https://www.jv-technoton.com/ru*](https://www.jv-technoton.com/ru)(дата обращения 04.04.2019).
8. Сайт АвтоГРАФ| Оборудование| Датчики уровня топлива — *URL:* [*https://glonassgps.com/dut*](https://glonassgps.com/dut)(дата обращения 04.04.2019).
9. Сайт *Omnicomm*| Продукты| Датчики — *URL:* [*https://www.omnicomm.ru/fuel-sensors*](https://www.omnicomm.ru/fuel-sensors)(дата обращения 04.04.2019).
10. Сайт *Gurtam*| Оборудование*|* Датчики — *URL*: [*https://gurtam.com/ru/sensors*](https://gurtam.com/ru/sensors)(дата обращения 04.04.2019).

1. Де-юре – формально. [↑](#footnote-ref-1)
2. Де-факто – означает нечто действительное, но не закрепленное в документации. [↑](#footnote-ref-2)
3. На рисунке 17 черным цветом – исходные данные, красным – сплайн без фильтрации, серым – первая производная. [↑](#footnote-ref-3)
4. На рисунке 10 черным цветом – исходные данные, красным – сплайн после фильтрации, серым – первая производная. [↑](#footnote-ref-4)
5. На рисунке 20 черным цветом – исходные данные, красным – сплайн, серым – первая производная, голубым – пороговое значение [↑](#footnote-ref-5)
6. На рисунке 21 черным цветом – исходные данные, красным – сплайн, серым – первая производная, голубым – пороговое значение. [↑](#footnote-ref-6)
7. На рисунке 22 черным цветом – исходные данные, красным – отфильтрованные значения. [↑](#footnote-ref-7)
8. На рисунке 23 черным цветом – исходные данные, красным – сплайн, розовым – первая производная. [↑](#footnote-ref-8)