**ДОКУМЕНТАЦИЯ на проект № 426**

1. **ТЕМА:** AutoGuard
2. **АВТОР:**

Тимон Стилиянов Трифонов

* ЕГН: 11.01.2006
* Адрес: гр. Враца,
* E-mail: [timon\_trifonov@abv.bg](mailto:timon_trifonov@abv.bg)
* Тел: +359876872050
* Училище: ППМГ „Акад. Иван Ценов“, гр. Враца
* Клас: XІ

1. **РЪКОВОДИТЕЛИ:**

Станислава Светославова Каменова

* Тел. 0879003288
* E-mail: [kamenova@bitex.bg](mailto:kamenova@bitex.bg)
* Старши учител по информатика и информационни технологии

Албена Кирилова Цекова

* Тел: 0879003176
* E-mail: [albena2008@abv.bg](mailto:albena2008@abv.bg)
* Старши учител по информатика и информационни технологии

**4. РЕЗЮМЕ:**

**4.1. Цел**

Приложение, представляващо цялостна система за откриване на аномалии, специално пригодена за автомобилната индустрия. Използвайки усъвършенствани техники за машинно обучение, включително дълбоко обучение и статистическо моделиране, системата анализира данни от различни сензори в двигателя на колата. Чрез този богат набор от данни, системата може да открива аномалии в реално или почти реално време, позволявайки активна поддръжка, предотвратяване на грешки и подобрени мерки за безопасност.

**4.2. Етапи в реализирането на проекта**

* Избор на тема и конкретна формулировка
* Преглед на съществуващи приложения
* Подбор на информация
* Създаване на подходящ дизайн
* Подбор на методи и средства за реализиране на идеята
* Изработване на авторски компоненти и материали
* Реализация
* Тестване на проекта
* Изработване на рекламни материали
* Представяне на проекта пред публика

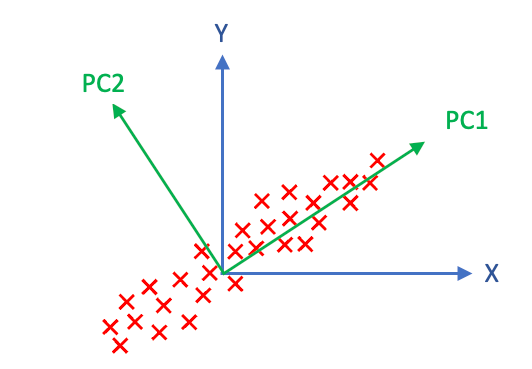
**4.3. Логическо и функционално описание на решението**

С нарастващата сложност на съвременните превозни средства и взаимосвързаността на техните системи, се появява необходимост от стабилни решения за откриване на аномалии. Тествани са различни алгоритмични решения на проблема, като в отговор на това търсене, е представен модел базиран на статистически алгоритъм. Приложението позволява на потребители без дори базови познания по машинно обучение да тренират и прилагат в реално време модела. При изполване на приложението потребителя най-напред се свързва с OBD (On-board diagnostics) устройство свързано към автомобила. То предава информация за двигателя към приложението. След това се създава модел или се избира по-рано създаден, който трябва да бъде трениран. След като приключи тренирането, моделът може да бъде избран за работа. В режим на работа(Inference mode) моделът връща процента аномалия при всяка извадка(sample) в реално време. Потребителя може да настройва големината на модела, кои параметри да използва при тренирането и праговата му стойност. Ако процента аномалия е по-голям от зададената прагова стойност, приложението записва аномалията и информация за нея в специален списък. Потребителя разглежда списъка и си прави изводи върху неговото съдържание, ако грешката е отстранена, то тя може да бъде изтрита.

**4.4. Реализация и ниво на сложност на проекта**

**Откриване на аномалии**

Алгоритъмът за откриване на аномалии е основно базиран на метода на главните елементи (МГЕ). Методът на главните компоненти е предимно техника за намаляване на размерността. Работи чрез идентифициране на основните компоненти. Основните компоненти са независими вектори на признаци, наричани също собствени вектори на дадени данни, които обясняват максималната дисперсия в данните. Всеки основен компонент е линейна комбинация от съществуващи корелирани характеристики и е ортогонален на други собствени вектори. С помощта на МГЕ можем да намалим броя на векторите на характеристиките, без да губим информационна стойност. Откриването на аномалия разчита на грешка при реконструкция. След като основните компоненти бъдат идентифицирани, чрез избиране на всички основни компоненти можем да реконструираме оригиналните данни от трансформираните данни без загуба. По същия начин, като избираме само основни компоненти, които обясняват по-голямата част от дисперсията, трябва да можем да пресъздадем приближение на оригиналните данни. Грешката, генерирана по време на реконструкция при генериране на оригиналните данни, се нарича грешка при реконструкция. За аномалии в данните грешката при реконструкцията е **висока**. За алгоритъмът са използвани Python и библиотеките PyOD, Numpy, Scikit-learn.

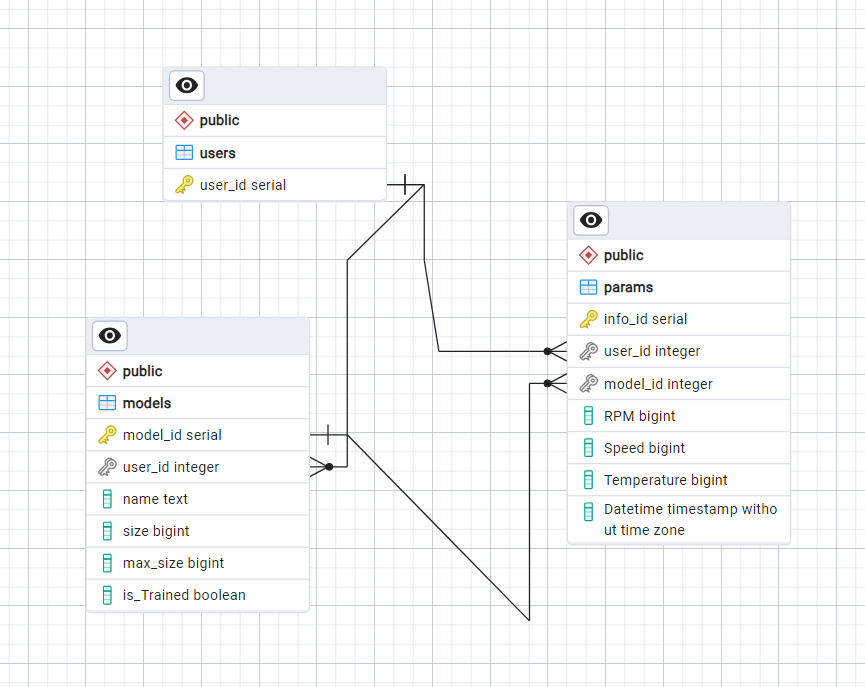


На изображението е представен примерен случай, където броят на характеристики е равен на две. За изчисляване на посоката на разпределението на данните(вектор PC1) се взима ковариационната матрица на извадката. Тук тя е

**База данни**

Базата данни е релационна и се състои от 4 таблици. В първата таблица се намира информация за потребителя, във втората събраните сензорни данни от автомобила, във третата- информация за съществуващи модели, създадени от потребителя, а във четвъртата се събира информация за всички върнати аномалии от модела. Разбира се и четирите са свързани по между си, посредством уникални и външни ключове. Базата данни е имплементирана чрез Postgressql. Връзката между приложението и базата данни се осъществява, чрез postgres.dart конектор модула.

(фиг. 1)



**Интерфейс**

Интерфейсът на мобилното приложение се състои от 3 раздела. (фиг. 2) Първият раздел (а) е начална страница с три бутона. Горе в дясно се намира бутон, който отваря менюто с настройки (б,в). То съдържа различни опции като свързване със bluetooth устройство, нотификации, промяна на език, помощ, но основната му функция е да дава на потребителя пълна свобода при избор на параметри, които модела да следи, неговата големина и праговата стойност, която ако бъде надвишена модела връща аномалии. В центъра на раздела са разположени 2 бутона. Първият води до менюто за събиране на данни и създаване и трениране на модели (г,д), втория бутон стартира така наречения Inference mode, който използва вече готов модел за работа в реално време. Втория (е,ж) раздел съдържа таблица с данните от сензорите на автомобила, както и графика отчитаща настоящия процент на аномалия. Третия раздел съдържа списък с всички засечени аномалии, подредени по дата и час, както и основна информация за тях. Долу в ляво са разположени 3 индикатора, 1 и 2 отчитат съответно връзката с интернет и сървъра, а 3 проверява съвместимостта на свързаното устройство и автомобила с приложението.

**Използвани технологии и модули :**

• Flutter, Dart, Bluetooth-serial - за разработка на интерфейса на мобилното приложение

• PostgresSQL - за базата данни

• Python, Sklearn, Numpy, Pyod, PythonOBD - за разработка на сървъра, модела за аномалии и тестване

**Тестване**

В по-голямата си част, приложението е тествано, чрез PythonOBD модула, който симулира сензорни данни от автомобил и ги изпраща на приложението, чрез Bluetooth.

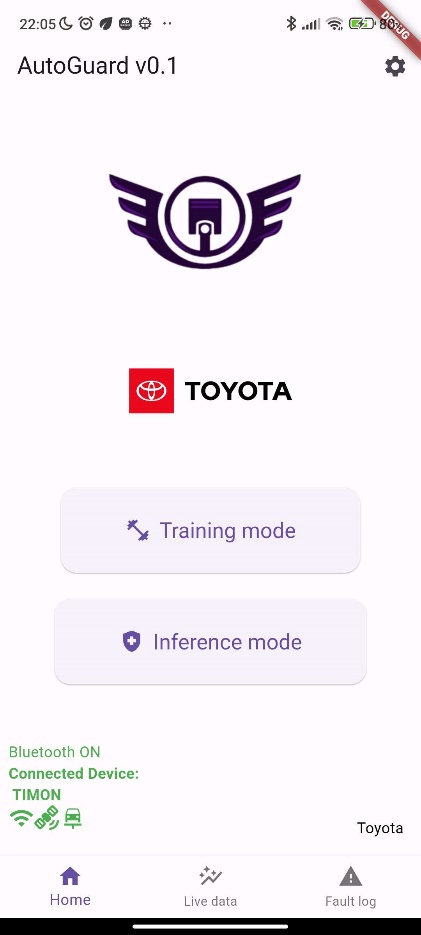
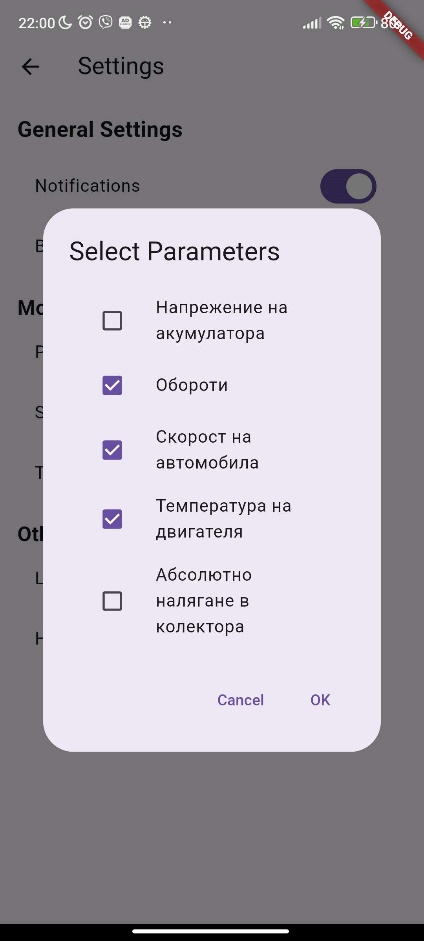
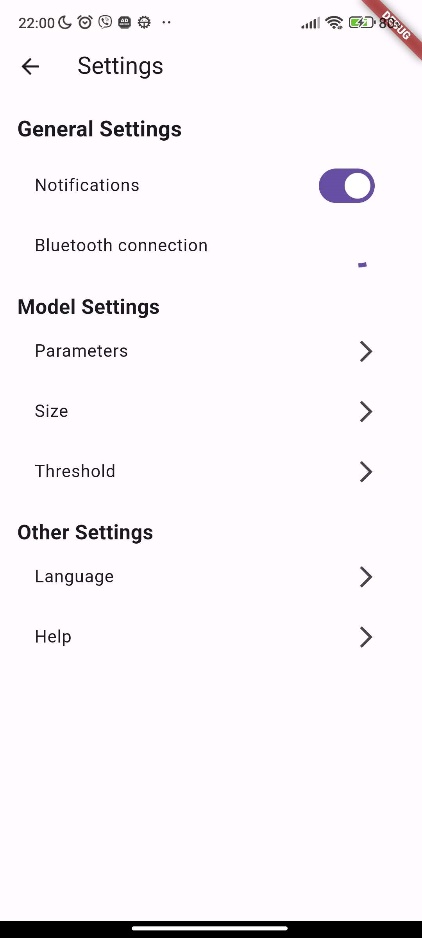
Проведени са и тестове в реални условия на автомобил Opel Meriva B5 с прикачено ELM327- OBD2 устройство. Приложението се справя почти перфектно, като единствените забележки са по-бавното събиране на данни от автомобила и периодичното подаване на невалидни данни от него.

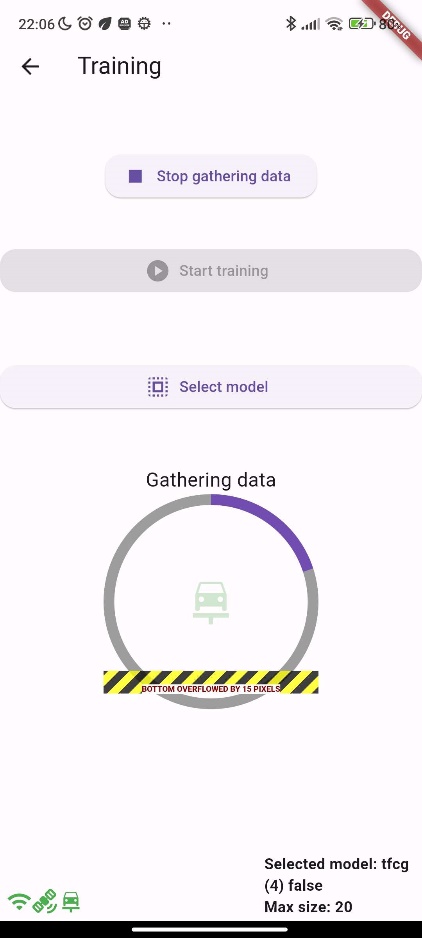
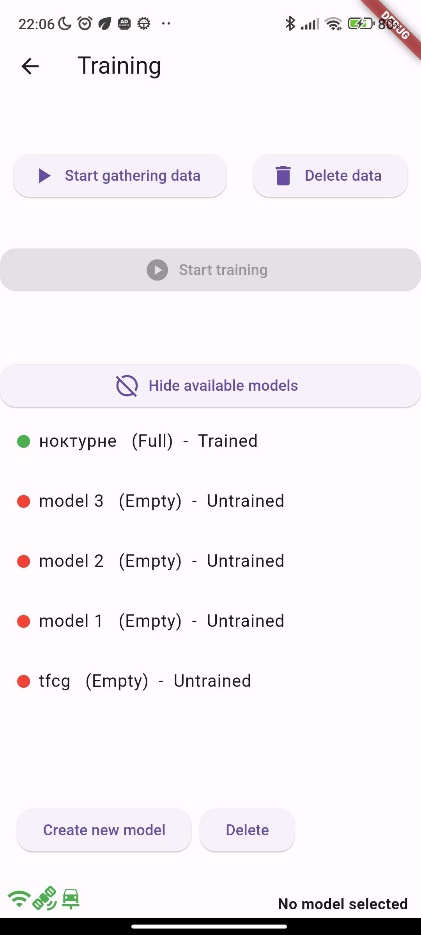
**4.5. Използвани източници**

* https://stackoverflow.com/
* <https://openai.com/blog/chatgpt/>
* Документация на Python, Sklearn, Numpy, Pyod, PythonOBD

**4.6. Изображения от проекта**

**(фиг. 2)**





**4.7. Описание**

Приложението се състои от няколко основни части, като връзката между автомобила и приложението се осъществява посредством OBD (On-board diagnostics) устройство, което предава данните от двигателя на външни приемници чрез bluetooth. Първата част осъществява връзката между приложението и OBD устройството. Тя събира и изпраща информация към OBD, като преди това провери за неговата достоверност и кодира(при изпращане) или декодира(при получаване) низа в шестнайсетична бройна система. Втората част е тази информация да бъде представена на потребителя чрез таблици и диаграми или да бъде записана в база данни за потенциално трениране на модел. Третата и последна част е използването на събраната информация за направата на завършен модел, който да бъде използван при откриването на аномалии.

Приложението е достъпно на адрес: <http://it.ppmg-vratsa.com/appgallery/>

**4.8. Заключение**

AutoGuard е приложение, което се справя успешно в задачата си да открива основни грешки в работата на двигателя на автомобила. То имплементира и разширява МГЕ алгоритъмът, като го пригодява за откриване на аномалии. Поради линейната същност на МГЕ, приложението не се справя добре при откриване на по-трудно забележими аномалии като някои контекстуални грешки, както и аномалии свързани с нелинейни процеси в работата на двигателя. Това е предпоствка за допълнителни изследвания и разширения по алгоритъма, като например kPCA – МГЕ алгоритъм, който използва така наречените кернел методи, придаващи нелинейност на модела.