



## **Система анализа эксплуатации оборудования с выявлением отклонений**

Булат Латыпов, Андрей Зозуля

Инжиниринг управления данными

AI Talent Hub, 2025

## **Краткое описание проекта**

### **Описание проекта:**

Проект направлен на разработку системы предиктивного технического обслуживания (ТО) с использованием методов машинного обучения для анализа данных, полученных с сенсоров оборудования. Система будет осуществлять мониторинг состояния оборудования в реальном времени и предсказывать необходимость проведения технического обслуживания на основе собранных данных.

### **Цели проекта:**

- Обеспечение автоматизированного мониторинга состояния оборудования.
- Выявление аномалий и прогнозирование необходимости ТО.

### **Методы и подходы:**

- Использование методов исследовательского анализа данных (EDA) для анализа собранных данных.
- Применение алгоритмов машинного обучения для обнаружения аномалий и изменения точек состояния (определения разладок процесса).
- Разработка MVP системы для тестирования и валидации подхода.

# Сбор данных

## Источники данных:

Данные (ток в Амперах) формируются электронными трансформаторами тока с выходом RS485, установленными на ремнях, нипроллерах и губках агрегатов фасовочных линий.

## Методы сбора:

Сбор данных непосредственно с датчиков по протоколу Modbus\* и частичная обработка (ModbusCollector).

- 1. Через фреймворк PyModbus\*\* производится подключение и опрос доступных устройств. Затем считывание с регистров датчиков в формате двух 16-битных чисел. Эти числа преобразуются в одно 32-битное.
- 2. Снятые значения отправляются в виде json на API сервиса (analyzer).

Загрузка данных из сегмента (ModbusCollector) на сервер проекта через API (analyzer), затем передача данных в базу данных ClickHouse\*\*\* через прокси Traefik\*\*\*\*.

## Пример данных:

| Timestamp          | Device                   | Amperage  |
|--------------------|--------------------------|-----------|
| 2024-12-01 0:00:01 | Токовый ремни Афл3.2     | 16.304153 |
| 2024-12-01 0:00:02 | Токовый ремни Афл3.3     | 3.9494066 |
| 2024-12-01 0:00:03 | Токовый нипроллер Афл3.3 | 0         |



\* Source: <https://www.modbus.org/> \*\* Source: <https://pymodbus.readthedocs.io/en/latest/> \*\*\* Source: <https://clickhouse.com/> \*\*\*\* Source: [Traefik Proxy Documentation - Traefik](#)

# Исследовательский анализ данных (EDA)

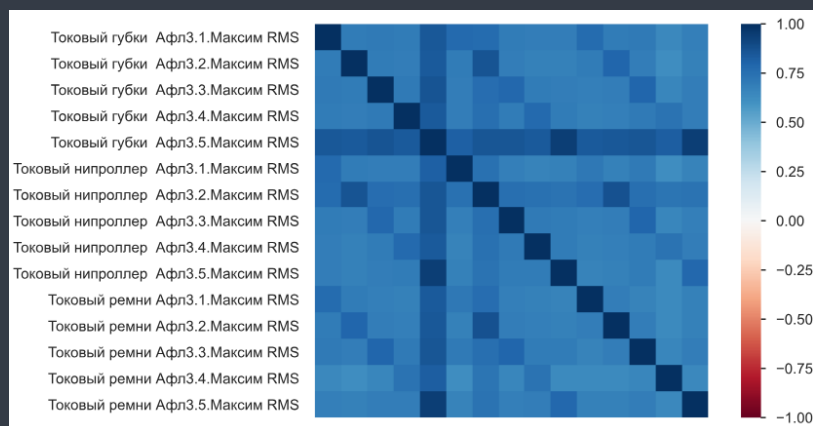
Для анализа данных использовалась библиотека YData Profiling\*  
Источник для EDA - выгрузки csv из базы данных ПО Simplight\*\*

Результаты приведены в основном для одного датчика (по остальным результаты схожие)

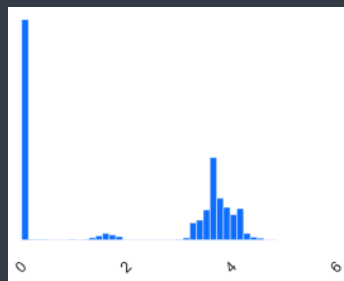
## Токовый ремни Афл3.1.Максим RMS

|                 |               |                |                |
|-----------------|---------------|----------------|----------------|
| <b>Distinct</b> | <b>276</b>    | <b>Minimum</b> | <b>0</b>       |
| Distinct (%)    | < 0.1%        | Maximum        | <b>178.5</b>   |
| Missing         | <b>610106</b> | Zeros          | <b>5865392</b> |
| Missing (%)     | <b>4.0%</b>   | Zeros (%)      | <b>38.6%</b>   |
| Infinite        | 0             | Negative       | 0              |
| Infinite (%)    | 0.0%          | Negative (%)   | 0.0%           |
| Mean            | 7.1562702     | Memory size    | 747.8 MiB      |

Обнаружено некоторое количество Nan значений (в нашем ETL они отсутствуют), и достаточно большое количество нулевых значений. Это связано с тем, что оборудование останавливается и запускается десятки раз в минуту. Много замерных значений выше номинального значения датчика.



Обнаружена корреляция между значениями токов на различных узлах агрегатов фасовочных линий.



На одном датчике может быть несколько распределений, что обусловлено различными настройками режимов работы оборудования.

\* Source: <https://docs.profiling.ydata.ai/> \*\* Source: <https://simplight.ru/>

## Метрики качества данных

### Доля значений, превышающих номинальное значение датчика (30 Ампер)

Для данной задачи это основная метрика определяющая качество данных и соответственно исправность датчиков, генерирующих эти данные

### Дополнительные метрики:

|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| Согласованность | корреляция между датчиками на одном агрегате должна быть более 0,7 (согласно EDA) |
| Актуальность    | время, прошедшее с момента последнего обновления данных.                          |
| Формат          | корректность формата даты, и количества знаков значения тока.                     |

Выявление аномалий и изменения трендов является целью проекта, поэтому выделены в отдельный слайд

# Разработка базы данных для хранения данных

ClickHouse — это колонно-ориентированная СУБД, что позволяет эффективно обрабатывать большие объемы данных и выполнять аналитические запросы с высокой скоростью. Это особенно важно для проектов, связанных с анализом данных генерируемых с датчиков.

## 1 Скрипт БД `init-db.sql` с следующими полями и упорядочиванием по дате:

|           |          |
|-----------|----------|
| timestamp | DateTime |
| device    | String   |
| amperage  | Float32  |

## 2 Докер контейнер

Создающий экземпляр БД ClickHouse с автоматическим созданием таблицы `sensor_data` (см. выше) при запуске контейнера.

## 3 Скрипт для тестирования

Формирующий запросы вида `"SELECT * FROM sensor_data"` и обрабатывающий исключения.



## Pipeline для автоматизации

**Сбор данных:** Используется протокол Modbus для сбора данных с датчиков (ModbusCollector).

Подключение и запросы:

- Использует фреймворк PyModbus для подключения и запроса доступных устройств с настройками IP-адреса, порта, адреса регистра, количества регистров.

Чтение и преобразование:

- Считывает регистры в два 16-битных значения и преобразуются в 32-битное число.
- Преобразование с плавающей точкой для точного представления данных.

Передача в Service API:

- Собранные значения отправляются в виде JSON в Service API (анализатор).
- Данные JSON включают временную метку, имя устройства и значения силы тока.

Логирование и обработка ошибок:

- Используется RotatingFileHandler для регистрации передачи данных.
- Реализованы предупреждения о недоступных устройствах и ошибки при неудачном чтении регистров.

**Загрузка данных:** Данные принимаются из ModbusCollector на сервер проекта через Service API.

Меры безопасности:

- Запросы API включают заголовки с ключом API для безопасности.
- Анализатор проверяет ключ API, чтобы предотвратить несанкционированный доступ.

Обработка данных:

- API анализатора получает и обрабатывает входящие данные.
- Используется CORS middleware для безопасных запросов между источниками.

Хранилище базы данных:

- Данные передаются в базу данных ClickHouse через прокси-сервер Traefik для эффективного хранения и извлечения.
- FastAPI обрабатывает маршрутизацию и предоставляет конечные точки для мониторинга и загрузки данных.

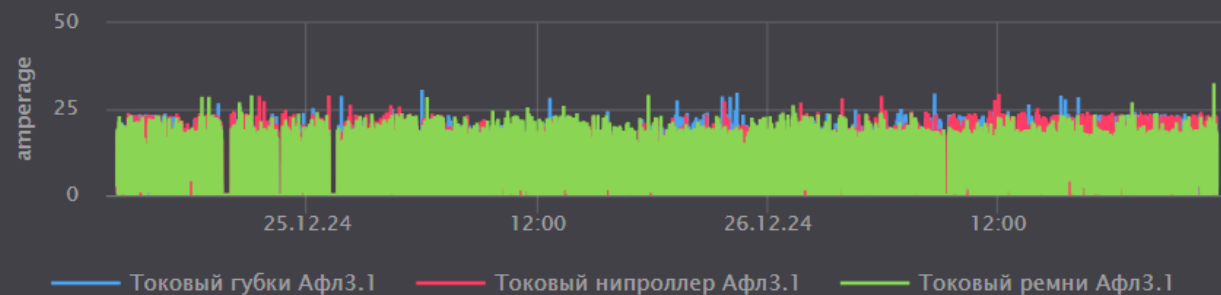


# Yandex Datalens для визуализации дашбордов

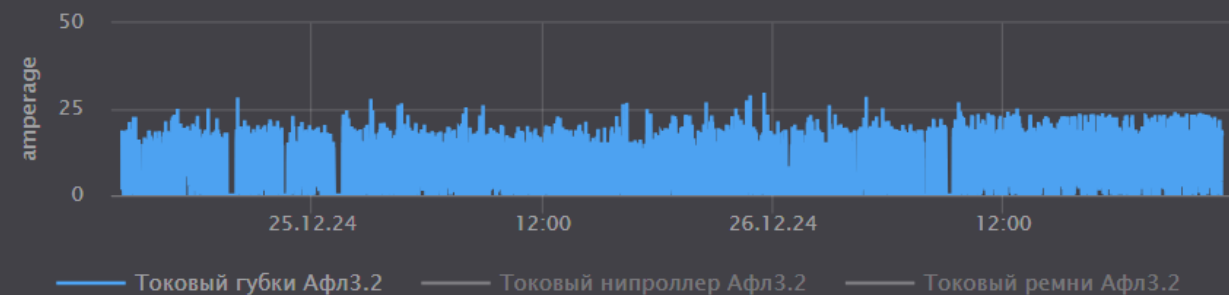
Ссылка на дашборд: <https://datalens.yandex/2rcp6ry0fx0go>

Визуализация данных:

Афл 3.1

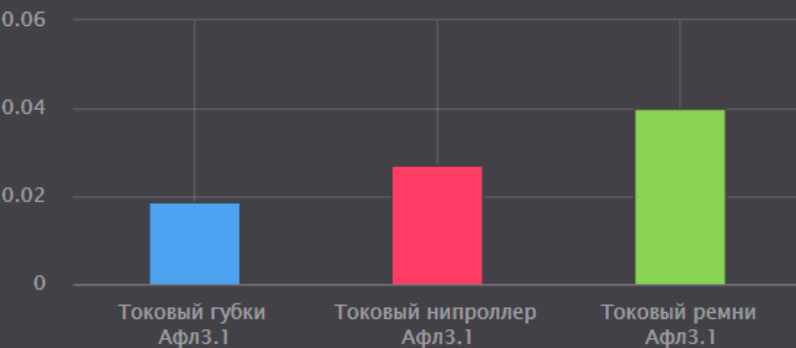


Афл 3.2

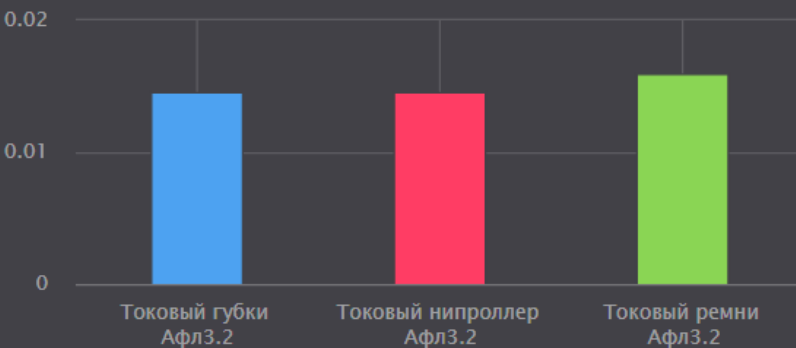


Визуализация метрик:

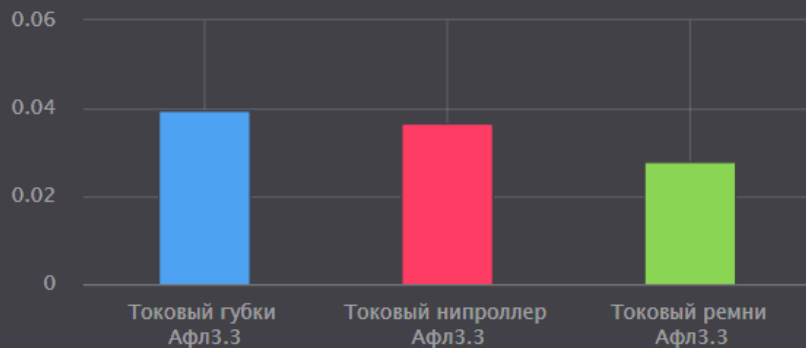
Афл 3.1 Доля значений превышающих номинальное



Афл 3.2 Доля значений превышающих номинальное



Афл 3.3 Доля значений превышающих номинальное





# Спасибо за внимание!

## Информация об авторах



Буллат Латыпов, магистрант 1 курса,  
<https://github.com/BulatMaratovich>  
@BulatMaratovich



Андрей Зозуля, магистрант 2 курса,  
<https://github.com/Anzovi>  
@Anzovitte