

# Тестовое задание От Бюро-1440

Шилов Артем Витальевич

Апрель 2025

## Задание 1: Анализ бейзлайн-решения

### Дано:

- Готовый код на Python (baseline.ipynb), который обнаруживает аномалии в данных датчиков двигателя (ток и температура).
- Результаты работы алгоритма: графики и метрики.

## Почему алгоритм плохо обнаруживает аномалии "Обрыв цепи" (резкое падение тока)?

Мне кажется дело в том что наш обрыв цепи происходит при токе равном примерно 0.1 А, что не выходит за рамки нашего распределения данных. А также алгоритм обучен одновременно на признаках по току и температуре, которая остается в норме в ходе данной аномалии. По итогу у нас нет особых отклонений признаков и в связи с этим у модели нет оснований для принятия обрыва цепи за аномалию.

## Как можно доработать алгоритм, чтобы улучшить детекцию?

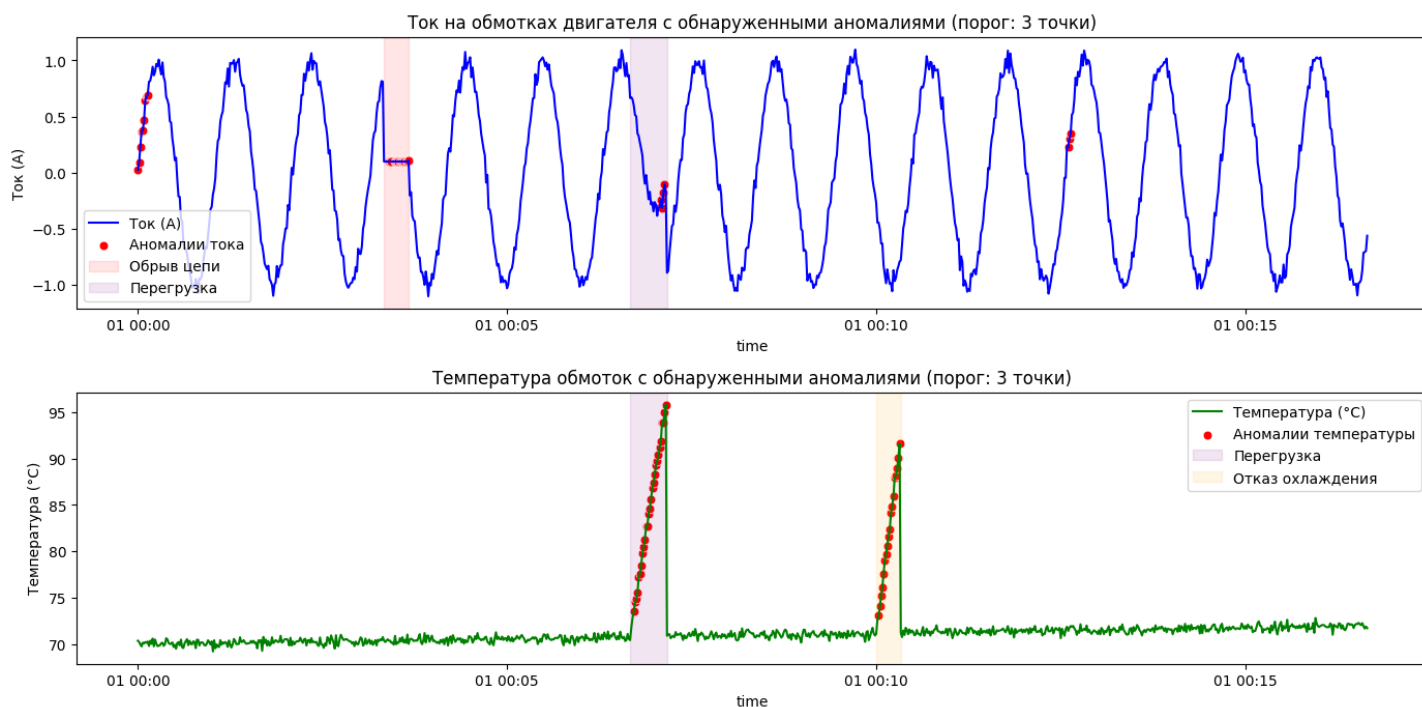
Изначально я попробовал добавить ещё параметр скорости изменения тока в Isolation forest, но это не принесло должных результатов т.к. алгоритм задетектил только ещё две добавочные аномалии в начале и в конце обрыва цепи. Тогда я решил попробовать обучить отдельно Isolation Forest для тока и для температуры с учетом моего прошлого апдейта (т.е. были добавлены фичи по скорости изменения тока на основе `df['current'].diff & df['current'].rolling`) и выбил вот такие метрики:

```
=== Эффективность детекции ===  
Обрыв цепи (200-220): 12/20  
Перегрузка (400-430): 27/30  
Отказ охлаждения (600-620): 18/20
```

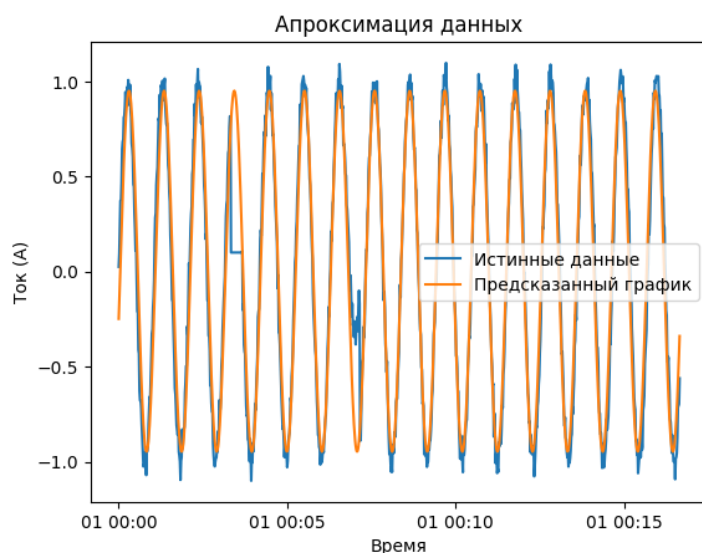
с учетом, что и для модели, которая работает с током, и с температурой были выставлены `contamination = 0.05`, а также аномалией считалась точка если хотя бы её признала таковой одна из моделей. Если же убрать `df['current'].diff` из фичей, то выбьется метрика чуть лучше, но ещё не совершенная:

```
=== Эффективность детекции ===  
Обрыв цепи (200-220): 14/20  
Перегрузка (400-430): 27/30  
Отказ охлаждения (600-620): 18/20
```

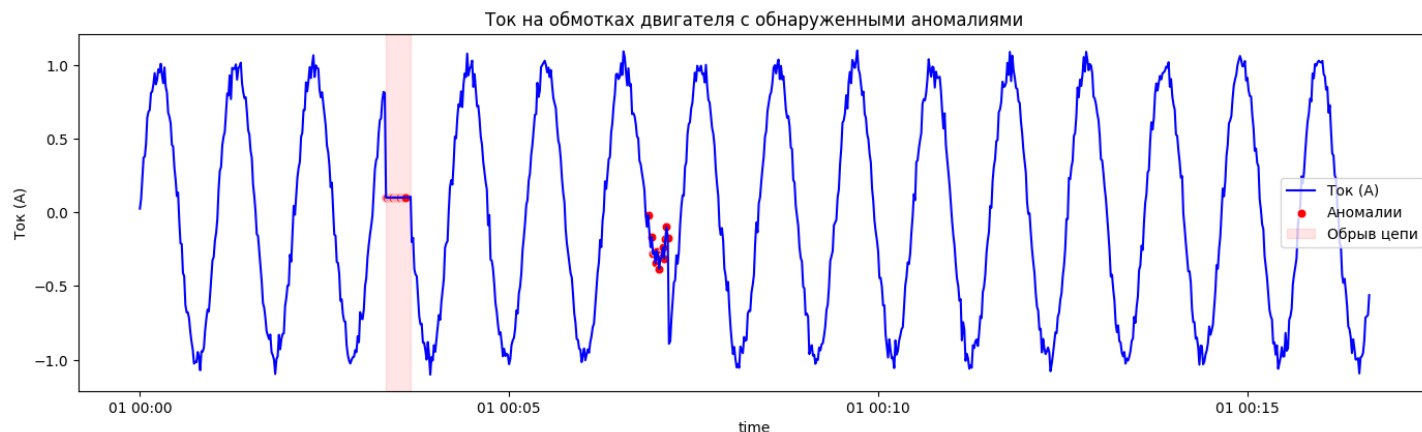
Итоговый график:



Далее, решил попробовать предложить другой алгоритм для обработки зависимости тока от времени. На основе данных было сделано предположения, что наши функции, описывающие данные, являются периодическими, а значит её можно разложить в ряд Фурье. На основе предположения применили быстрое преобразование фурье (FFT) и линейную регрессию для нахождения коэффициентов ряда. А затем можно найти разницу между истинной функцией и аппроксимированной нами:



Выбрав трэшхолд, можно детектировать аномалии в случаях, когда разница между истинной и предсказанной функцией его превышает. По итогу получилим метрику детекцию 17/20 обрыва цепи, что является наилучшим результатом:



## Как уменьшить количество ложных срабатываний?

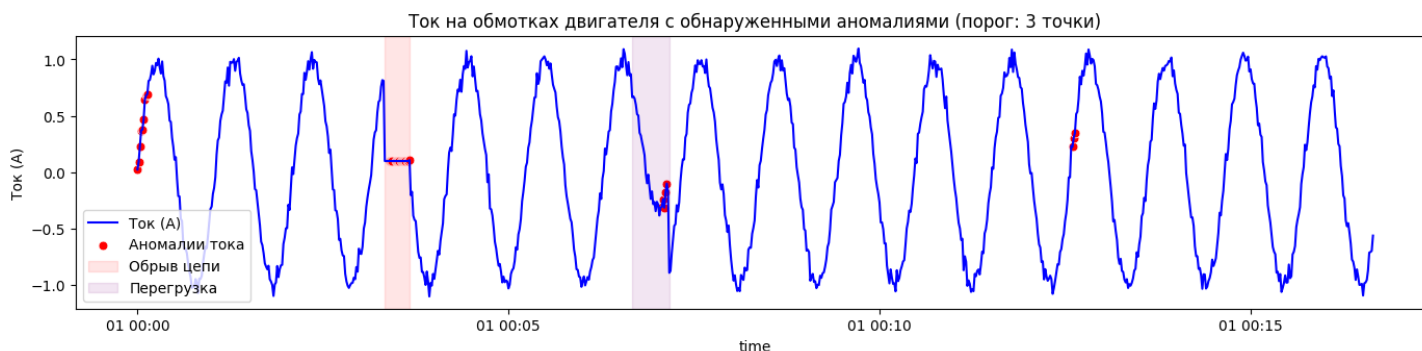
Давайте для начала введем метрику Precicsion по которому будем смотреть качество модели. Посчитаем метрику для наших моделей:

```
Считаем точность:  
Точность для тока: 0.50  
Точность для температуры: 0.90  
Объединенная точность: 0.71
```

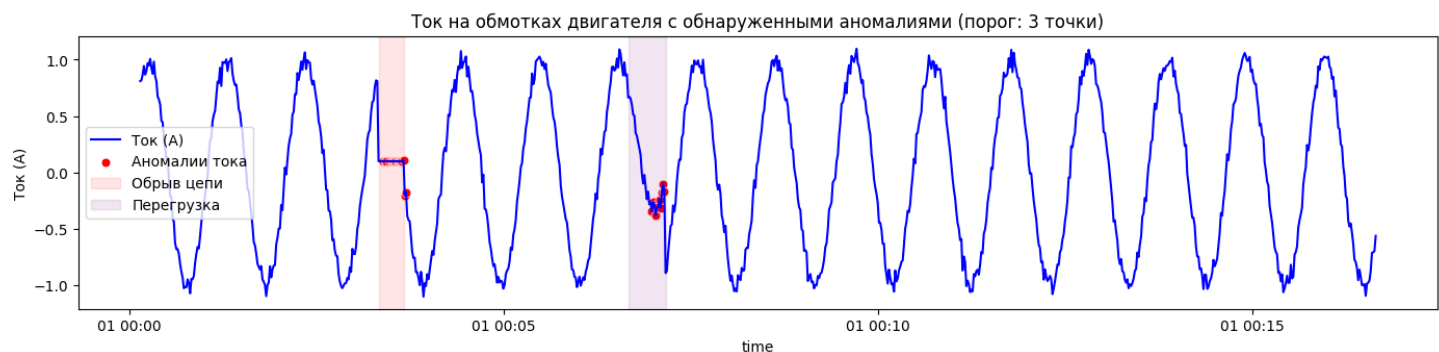
Отсюда можно заметить, что минимизирует нашу общую точность модель, работающая с током, поэтому давайте её максимизировать. Первым делом давайте поставим порог по количеству подряд срабатываний, чтобы принять их за реальную аномалию. С данным алдейтом мы получили метрики:

```
Считаем точность:  
Точность для тока: 0.62  
Точность для температуры: 1.00  
Объединенная точность: 0.83
```

Давайте посмотрим наш график с аномалиями и заметим, что у нас детектируются первые точки из-за того что фици rolling-mean и rolling-std, связанные с током, начинают использоваться только после 9 сэмпла:



Поэтому давайте уберем объекты, где имеются NaN значение т.е. первые 9. Тем более мы не сможет выявить на первых значениях аномалию т.к. не знаем как себя истинно ведут показатели и получим:



Итоговые метрики будут:

```
Считаем точность:  
Точность для тока: 0.93  
Точность для температуры: 1.00  
Объединенная точность: 0.97
```

Код будет приложен в сообщении!

## Задание 2: Размышления о задаче в реальных условиях

### Контекст:

В реальных промышленных системах данные сложнее:

- Больше параметров (вибрация, давление, скорость вращения).
- Неочевидные зависимости (например, перегрев может проявляться с задержкой).
- Нет точной разметки аномалий.

### Как оценивать качество обученной модели?

Начнём с того, что за самую верхнеуровневую метрику можно взять кол-во аварий которые смогли предотвратить за счёт модели. Говоря о более низкоуровневых метриках, мы можем посмотреть качество на отложенной выборке или посмотреть кол-во найденных аномалий, размеченных экспертами

#### Более точно:

- **Классические метрики (если есть разметка):** Если нам известны хотя бы приблизительные интервалы аномалий (например, по записям инцидентов), можем попробовать классические метрики классификации: Precision, Recall (в нашей задаче стоит уделить особое внимание этой метрике, так как высока цена ошибки), F1-score и тд. Тут можем применить несколько трюков, к примеру: обнаружение хотя бы одной точки внутри аномального интервала засчитывается как обнаружение всей аномалии
- **Специфичные метрики для временных рядов:** Исходя из соображения, что нам надо моментально детектировать аномалии, имеет смысл попробовать метрики, учитывающие задержку обнаружения и длительность аномалий. Например, NAB score (метрика бенчмарка Numenta) [Подробнее: <https://habr.com/ru/companies/rosatom/articles/687270/>] специально вознаграждает более раннее обнаружение аномалии и штрафует за опоздание, ложные срабатывания и пропуски

- **Сами искусственно проверять модель:** Подобно тестовому заданию, мы можем искусственно добавлять аномалии в данные для тестирования. К примеру, добавлять в тестовые данные всплески, выбросы, искажение сигналов, имитирующие реальные неисправности, и проверять, нашла ли их модель

Во время работы модели, также можем смотреть за поведением модели. Точно имеет смысл отслеживать среднее время до обнаружения (MTTD) – время от начала реальной неисправности до срабатывания алгоритма, а также количество ложных тревог в неделю/месяц и процент выявленных известных инцидентов

## Можно ли применить обучение с учителем? Если да, то как разметить данные?

Применение полностью supervised обучения для аномалий затруднено из-за дефицита достоверной разметки. Как я понял, в реальных промышленных данных крайне мало заранее помеченных примеров всех видов неисправностей. В том числе потому что появляются новые типы аномалий, не встречавшиеся ранее, поэтому полная разметка “на все случаи” практически недостижима. Отсюда сразу идея: **не обучать модель находить аномалии разных типов, а обучить её предсказывать "всё ли идёт как обычно тогда мы получаем обычную бинарную классификацию, и все случаи, отклоняющиеся от нормальной работы будут зафиксированны как аномалии.**

Различные методы:

- **Разметка нормальных данных:** Более реально пометить периоды нормальной работы. Если известны интервалы, когда система работала без сбоев, их можно пометить как “норма”. Однако, поскольку датчики в разных условиях выдают разные результаты, и все они считаются нормой, можем сначала классифицировать в каких условиях мы находимся, а затем решать задачу бинарной классификации. Тогда задача сводится к one-class classification – обучается модель нормального поведения, а отклонения от него считаются аномалией (уже про это упомянули)
- **Частичная разметка аномалий:** Если доступно несколько примеров аномальных случаев (пусть даже неточно локализованных), их можно использовать для улучшения модели (можно использовать contrastive learning)
- **Синтетические данные:** Как в прошлом пункте, можно создавать симуляции и генерировать аварийные / аномальные ситуации.

## Какие современные методы (2023-2025) стоит попробовать и почему?

Я решил углубиться какие методы в AD являются актуальными в наши дни. Для того чтобы узнать в какую сторону идти я ознакомился со статьей: "Anomaly Detection Methods for Industrial Applications: A Comparative Study" в которой сверяются supervised и unsupervised подходы, а именно классическое ML с ручным извлечением признаков и DL методы с автоэнкодерами. В статье на основе метрик, делают вывод, что DL методы работают лучше и автоматически извлекают признаки из сырых данных, поэтому будем углубляться именно в этот подход.

### Подход на основе отклонения значения

Одним из базовых подходов к решению задачи аномалии является обучение рекуррентных нейросетей (LSTM, GRU, LSTM-AE (LSTM с автоэнкодером)) или авторегрессионных моделей (ARIMA). Суть метода в том чтобы считать отклонение между предсказанным и истинным значением и детектировать аномалию в случае превышения трэшхолда. Главным плюсом является в силу небольшого времени для инференса, возможность использовать их в онлайн режиме для успешного предотвращения поломок. Например: LSTM-AE модель была внедрена прямо на микроконтроллер для мониторинга

вибрации двигателя. Её размер составил всего 22 KB, при этом она выявляла аномалии с точностью около 82% [Ссылка на статью: <https://www.itrvn.com/case-studies/advancing-predictive-maintenance>] Но есть проблема в том что она хорошо работает только с короткими аномалиями.

Генеративные модели

В дальнейшем резерче я наткнулся на статью: "TSI-GAN: Unsupervised Time Series Anomaly Detection using Convolutional Cycle-Consistent Generative Adversarial Networks"в которой сравниваются модели для детекции аномалий в том числе на аэрокосмических данных NASA. Вот метрики из статьи:

Table 3: Average F1-Score on original and distorted datasets for each category, as well

Model	AirTemperature		PowerDemand		InternalBleeding		EPG		NASA T-1		Noise	All 250 datasets	
	Orig.	Distor.	Orig.	Distor.	Orig.	Distor.	Orig.	Distor.	Orig.	Distor.		F1	Precision
TSI-GAN	1.0	0.833	0.667	0.667	0.846	0.474	0.5	0.556	0.933	0.267	0.479	0.468	0.445
MERLIN	0.054	0.18	0.04	0.071	0.926	0.721	0.354	0.191	0.613	0.6	0.49	0.414	0.402
LSTM-AE	0.389	0.611	0.375	0.583	0.654	0.308	0.222	0.444	0.533	0.333	0.208	0.355	0.301
DONUT	0.611	0.444	0.083	0.1	0.59	0.564	0.278	0.167	0.333	0.533	0.458	0.351	0.325
LSTM-DT	0.778	0.833	0.25	0.5	0.615	0.449	0.222	0.222	0.6	0.6	0.271	0.32	0.289
DENSE-AE	0.194	0.111	0.0	0.0	0.231	0.077	0.222	0.222	0.2	0.0	0.271	0.159	0.136
TadGAN	0.0	0.133	0.0	0.0	0.282	0.24	0.233	0.189	0.267	0.2	0.171	0.131	0.092
Azure	0.181	0.199	0.083	0.196	0.099	0.176	0.167	0.167	0.007	0.017	0.084	0.05	0.037
Luminol	0.022	0.021	0.078	0.089	0.118	0.046	0.037	0.088	0.009	0.014	0.019	0.049	0.021

Table 3: Average F1-Score on original and distorted datasets for each category, as well

Поэтому можно рассмотреть её. [Ссылка на статью: <https://arxiv.org/abs/2303.12952>]

Модели на основе Transformers:

В контексте аномалий, attention-слой означает, что модель может сопоставлять текущие показания не только с недавней историей, но и с типичными паттернами за долгий период (таким образом решая проблему LSTM И находя долгие отклонения от нормы). Пример – Anomaly Transformer, который ввёл специальный anomaly-attention для различения нормальных и аномальных точек по их «ассоциативным связям» с остальной серией [Ссылка на статью: <https://arxiv.org/abs/2110.02642>] Идея заключается в том, что у нормального сигнала каждая точка связана с другими (например, сезонными или коррелирующими показаниями), а у редкой аномалии связи нарушаются и внимание модели фокусируется лишь на ближайших соседях.