Поиск паттернов в многомерных временных рядах для предотвращения отказов в технологическом оборудовании

Описание задачи

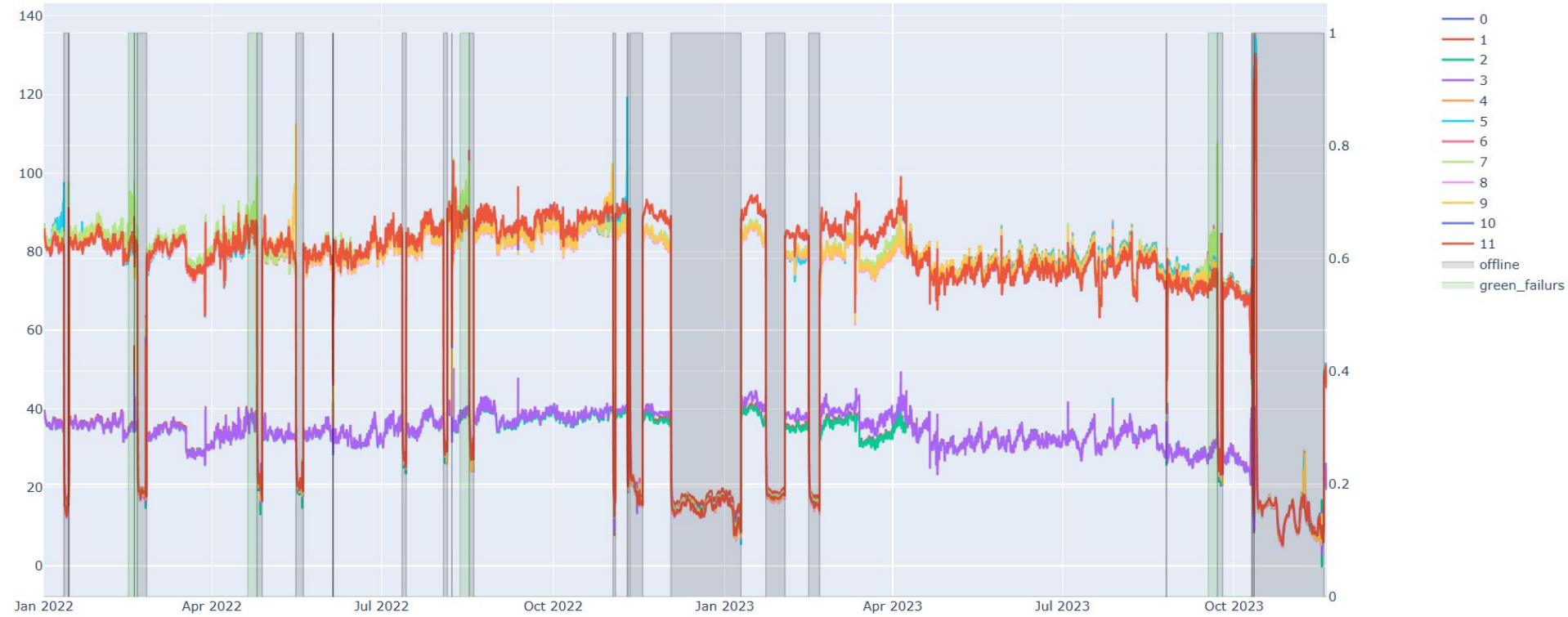
Цель - разработать модель для поиска в многомерных временных рядах паттернов, предшествующих отказу технологического оборудования.

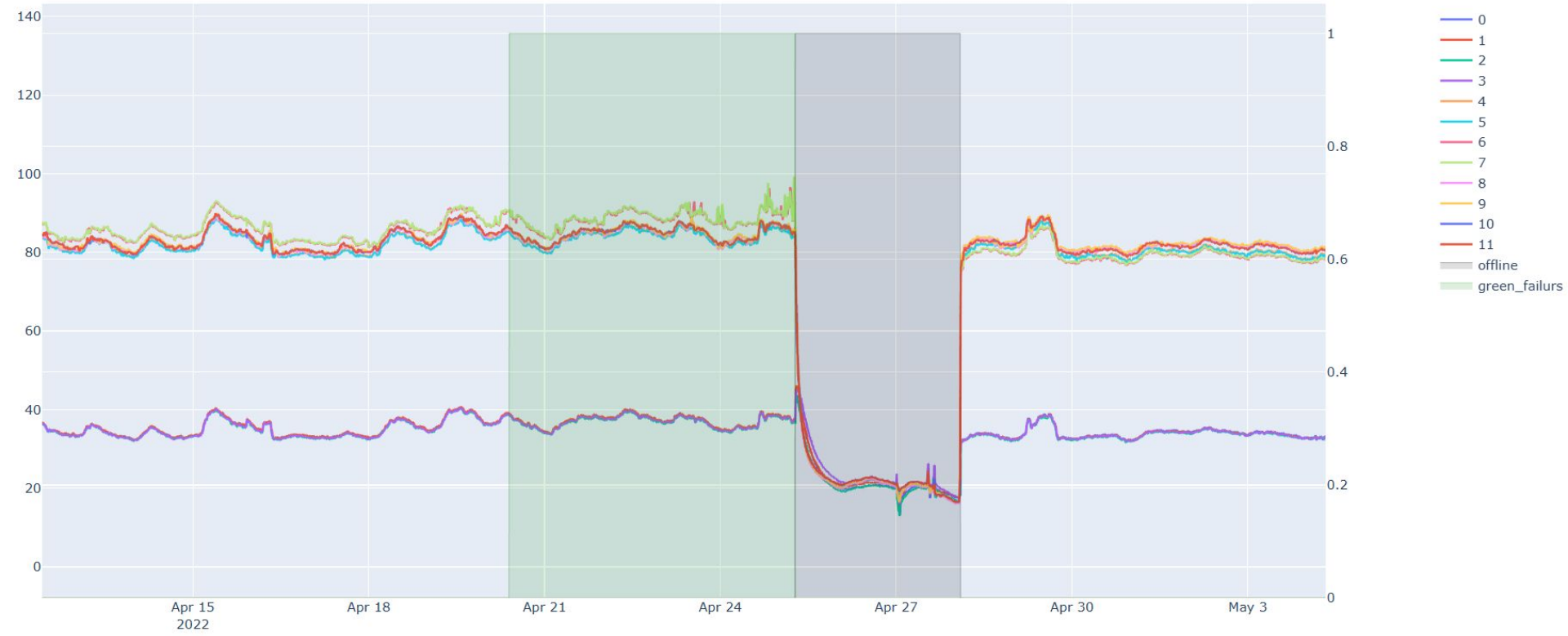
Задача классификации многомерных временных рядов.

Данные - показания с датчиков температуры подшипников компрессора за 2 года.

1 EDA

- 12 столбцов, 99 тыс. временных меток;
- Пропусков и дубликатов нет;
- Данные упорядочены и ресемплированы по 10 минут;
- В данных есть участки, когда оборудование не работало - офлайн-периоды (18 % от всех данных). Они не будут использоваться в обучении и тестировании;
- В данных есть 4 участка отказа с одинаковым паттерном.

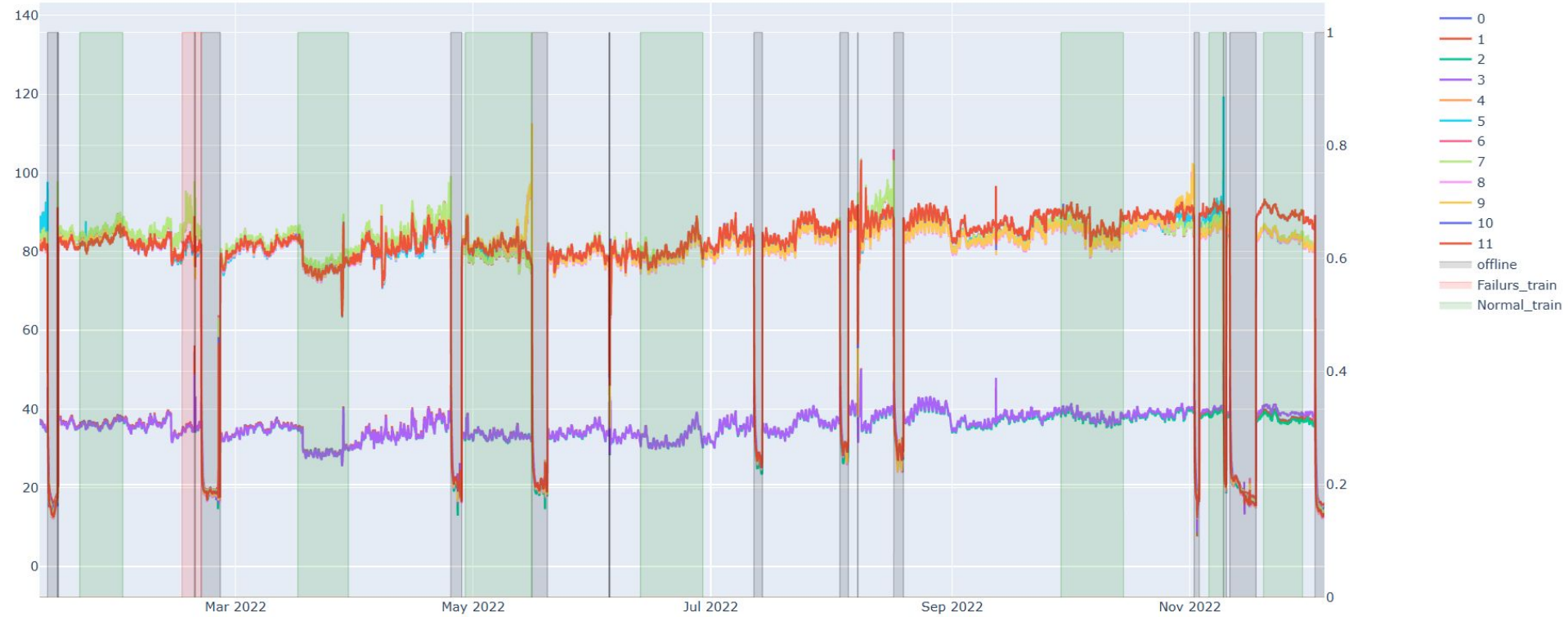

140
120
100
80
60
40
20
0
1
0.8
0.6
0.4
0.2
0
Jan 2022
Apr 2022
Jul 2022
Oct 2022
Jan 2023
Apr 2023
Jul 2023
Oct 2023
0
1
2
3
4
5
6
7
8
9
10
11
offline
green_failurs


140
120
100
80
60
40
20
0
1
0.8
0.6
0.4
0.2
0
Apr 15
2022
Apr 18
Apr 21
Apr 24
Apr 27
Apr 30
May 3
0
1
2
3
4
5
6
7
8
9
10
11
offline
green_failurs

2 Preprocessing - train

- Для обучения - участки данных только за 2022 год
- Для обучения - только первый отказ.
- Каждый отказ размечается периодом за 5 дней до отключения оборудования.
- Каждый объект – окно данных длительностью 8 часов (48 точек)
- Обучающая выборка: 13 тыс. объектов, 5 % - класс 1, 95 % - класс 0

Разметка обучающей выборки



Preprocessing - test

- Для тестирования используются все данные, кроме офлайн-участков
- Метрики: f1-score, MCC
- 3 участка отказа длительностью 5 дней каждый
- Участки нормальной работы – все остальные участки с отступом от офлайн-участков 1 день и от размеченных отказов – 15 дней
- Тестовая выборка: 56 тыс. объектов, 4 % - класс 1, 96 % - класс 0

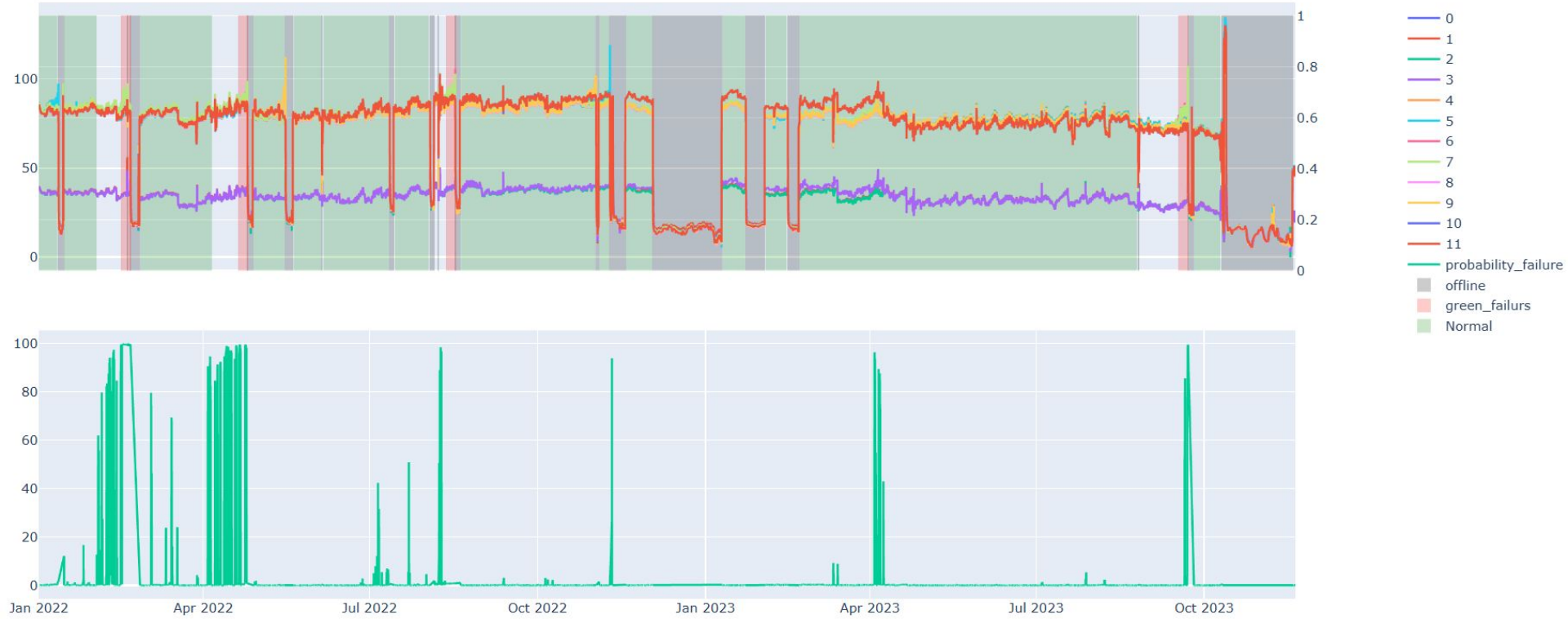
3 Modelling

- Tsfresf + CatBoost
- CNN
- Vanilla Transformer
- CNN + Transformer
- Informer
- iTransformer

3.1 Tsfresf + CatBoost

- Немасштабированные данные
- Генерируется 120 признаков
- Учет дисбаланса – взвешивание
- ROC-AUC = 0,5604
- f1 = 0,2
- MCC = 0,2331

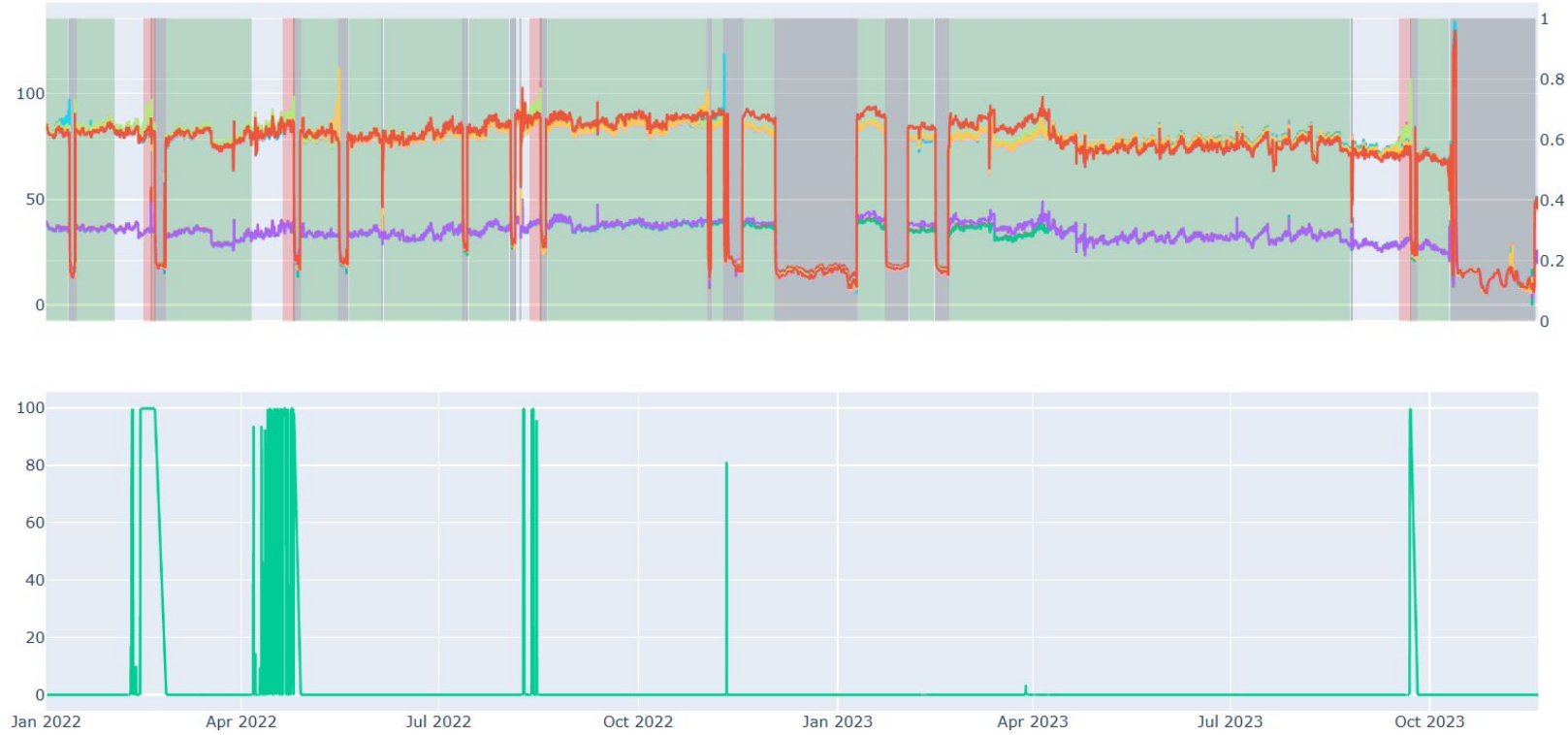
Tsfresf + CatBoost



3.2 CNN

- Масштабирование - StandardScaler
- Слои: 3 conv1d + 2 fc
- epochs = 20
- batch_size = 32
- Optimizer - Nadam(learning_rate=0.001)
- ROC-AUC = 0,616
- f1 = 0,375
- MCC = 0,468

CNN


100
50
0
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10
11
probability_failure
offline
green_failurs
Normal
100
80
60
40
20
0
Jan 2022
Apr 2022
Jul 2022
Oct 2022
Jan 2023
Apr 2023
Jul 2023
Oct 2023

CNN biased


100
50
0
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10
11
probability_failure
offline
green_failurs
Normal
100
80
60
40
20
0
Jan 2022
Apr 2022
Jul 2022
Oct 2022
Jan 2023
Apr 2023
Jul 2023
Oct 2023

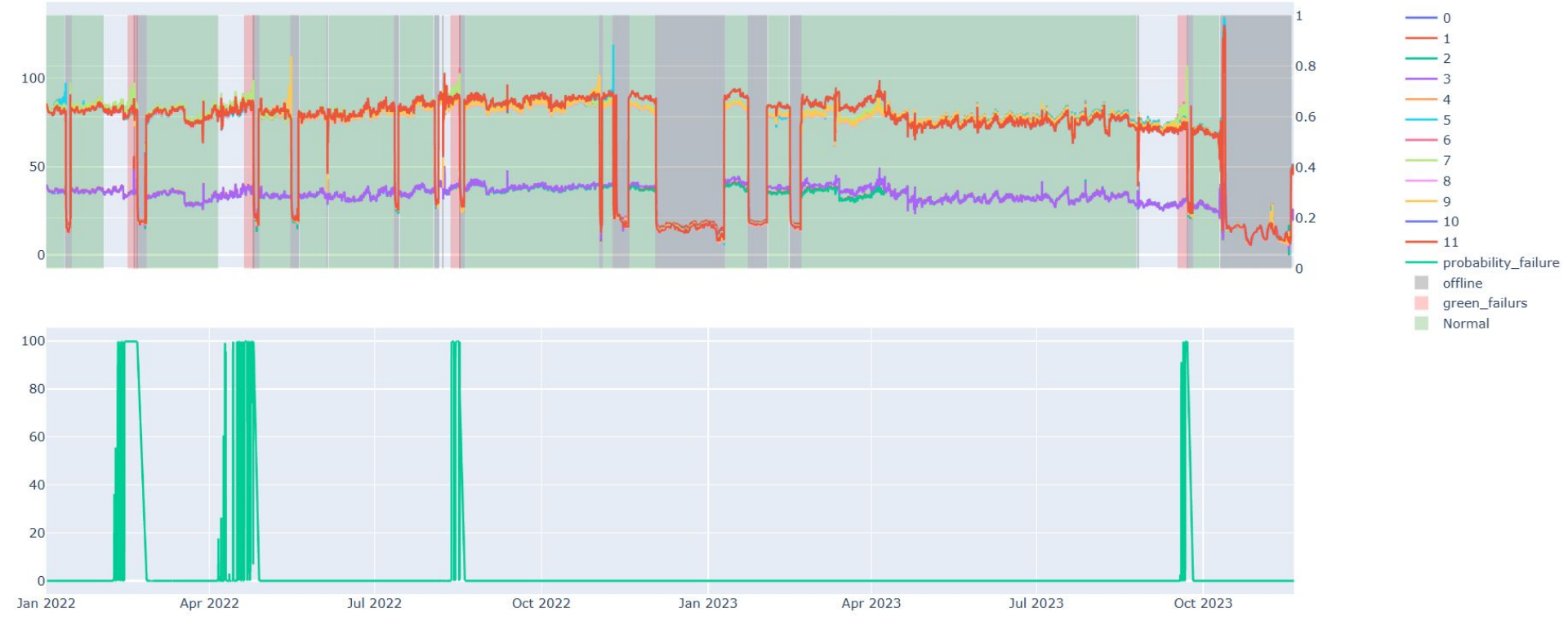
CNN win

- Масштабирование каждого окна:

$$\bar{x}'_{win,tag} = \bar{x}_{win,tag} - \mu_{win,tag}, \qquad win = \overline{1, N_{win}}, tag = \overline{1, N_{tag}}$$

- ROC-AUC = 0,881
- f1 = 0,864
- MCC = 0,868

CNN win


100
50
0
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10
11
probability_failure
offline
green_failurs
Normal
100
80
60
40
20
0
Jan 2022
Apr 2022
Jul 2022
Oct 2022
Jan 2023
Apr 2023
Jul 2023
Oct 2023

3.3 Vanilla Transformer

- Слои: 1 fc + 2 TransformerEncoder + 1 fc
- TransformerEncoder – кодирование временных меток
- epochs = 20
- batch_size = 32
- Optimizer - Adam(learning_rate=0.001)

	ROC-AUC	f1	MCC
StandardScaler	0,6643	0,4948	0,5658
win	0,9833	0,9830	0,9825

Vanilla Transformer win

3.4 CNN + Transformer

- Слои: 2 conv1d + 4 TransformerEncoder + 2 fc
- TransformerEncoder – кодирование временных меток
- epochs = 10
- batch_size = 32
- Optimizer - Adam(learning_rate=0.001)

	ROC-AUC	f1	MCC
StandardScaler	0,5468	0,1713	0,3006
win	0,9918	0,9888	0,9884

CNN + Transformer win

3.5 Informer

- Модифицированный механизм внимания
- Слои: 1 fc + 4 TransformerEncoder + 1 fc
- epochs = 20
- batch_size = 32
- Optimizer - Adam(learning_rate=0.001)

	ROC-AUC	f1	MCC
StandardScaler	0,6564	0,4759	0,5497
win	0,9868	0,9866	0,9862

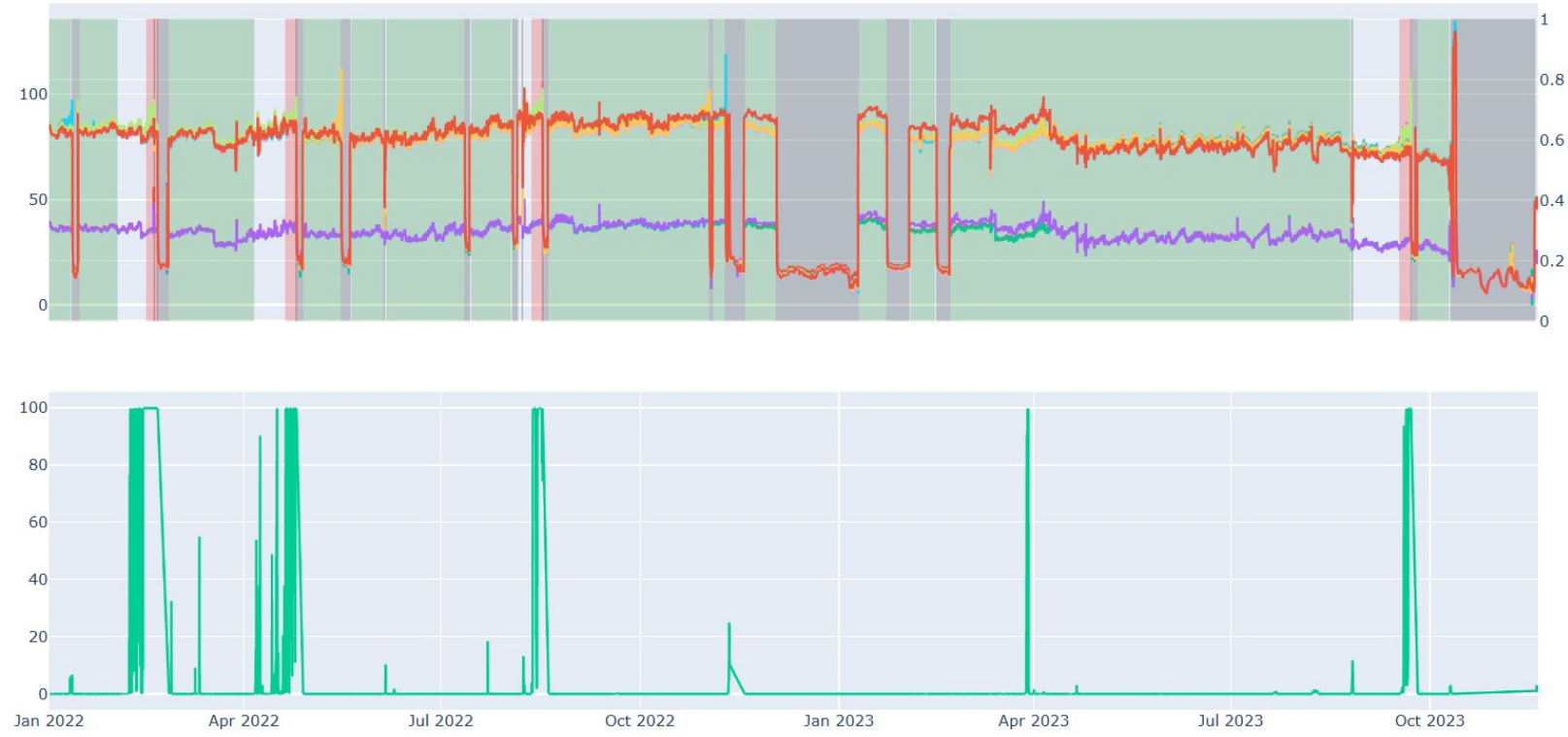
Informer win

3.6 iTransformer

- Inverted TransformerEncoder – кодирование признаков
- Слои: 1 fc + 4 TransformerEncoder + 2 fc
- epochs = 20
- batch_size = 32
- Optimizer - Adam(learning_rate=0.001)

	ROC-AUC	f1	MCC
StandardScaler	0,6963	0,5516	0,5929
win	0,6935	0,5267	0,5507

iTransformer win



	ROC-AUC	f1	MCC
cnn_tr_win	0.991766	0.988816	0.988385
informer_win	0.986787	0.986610	0.986177
tr_win	0.983310	0.983027	0.982512
cnn_win	0.880621	0.864438	0.868353
itr	0.696337	0.551590	0.592896
itr_win	0.693546	0.526710	0.550744
tr	0.664349	0.494766	0.565771
informer	0.656421	0.475855	0.549699
cnn	0.616004	0.375000	0.467742
tsfresh_cb	0.560360	0.200074	0.233080
cnn_tr	0.546824	0.171259	0.300617

4 Выводы

- Масштабирование каждого окна путем приведения к нулю среднего на окне для каждого тега значительно улучшает результаты всех протестированных моделей, кроме iTransformer
- Результаты алгоритма iTransformer с масштабированием каждого окна и слабо отличаются
- Худший результат по метрикам ROC-AUC и f1 показала модель CNN + Transformer, а по метрике MCC - tsfresh + CatBoost
- Наилучшая модель по всем метрикам - CNN + Transformer с масштабированием каждого окна