**ДИПЛОМНАЯ РАБОТА**

**Выявление аномалий в данных на ветроэнергетических установках**

Направление: Data Scientist

Группа: DS-75

Студентка \_\_\_\_\_\_\_\_\_\_ Волкова Н.С.

# Содержание

[1. Введение и постановка задачи 3](#_Toc171594623)

[2. Описание данных и их особенностей 4](#_Toc171594624)

[3. Описание обработки данных и разделение на обучающие и тестовые данные 9](#_Toc171594625)

[4. Описание решения и архитектура 11](#_Toc171594626)

[5. Описание обучения 15](#_Toc171594627)

[6. Описание итогового результата 20](#_Toc171594628)

[7. Заключение с выводами и планами на дальнейшее развитие 23](#_Toc171594629)

[8. Источники, использованные при разработке 24](#_Toc171594630)

# 1. Введение и постановка задачи

На ветряной станции имеются ветроэнергетические установки, с помощью специального программного обеспечения с каждой из этих установок через определённые промежутки времени считываются данные о вибрации. Полученные данные сохраняются в облачном хранилище.

Со временем показания могут меняться, что может свидетельствовать о неполадках в работе установки.

Задача исследования — обучить нейронную сеть анализировать данные и определять, находилась ли конкретная установка в заданный промежуток времени в состоянии, отличном от нормального. Для этого нейронная сеть будет использовать свой предыдущий опыт и информацию о показаниях вибрации.

Данная разработка будет полезна операторам АРМов (АРМ – автоматизированное рабочее место) на ветряных электростанциях для оперативного выявления появившихся неисправностей в работе ветроэнергетических установок и предотвращении аварийных остановок.

# 2. Описание данных и их особенностей

Данные хранятся в облачной базе данных InfluxDB в таблице «stat».

На каждом из агрегатов установлено по 8 датчиков. Каждый датчик измеряет три вида данных с разными интервалами времени. Эти данные хранятся в типе данных float64.

- **FILTER\_SENSOR** — фильтрованные данные;

- **HIGH\_SENSOR** — высокочастотные (ВЧ) данные;

- **LOW\_SENSOR** — низкочастотные (НЧ) данные.

Для каждого датчика и вида данных рассчитываются следующие показатели:

- **crest**, **skew**, **kurtosis** — Крест-фактор, Перекос, Куртозис, соответственно. Они присутствуют в базе, но в аналитике они не используются, т. к. они вычислены упрощенно и есть другие, более показательные данные;

- **fband** — среднеквадратичные значения (СКЗ) диапазонов частот, указанные в документации производителя;

- **peak2peak**, **peak** — разница между минимумом и максимумом;

- **rms** — среднеквадратичные значения;

- **so\_hs\_is**, **so\_hss**, **so\_iss**, **so\_lss** — СКЗ частот валов, кратных оборотным, от одного до трёх.

Кроме того, для каждого измерения доступна следующая информация:

- **tgnum** — наименование агрегата (WTG1, WTG2 и так далее). Тип данных: object.

- **warning** — «0» или «1». «0» — условно нормальный режим (существующая система не выдала предупреждение), «1» — режим с предупреждением. Этот показатель не является актуальным на текущем этапе работы программного обеспечения. Тип данных: object.

- **plant** — код станции (W1436), одинаковый для всех в нашем случае. Тип данных: object.

Описание низкочастотных данных с агрегата «WTG1» датчика «SENSOR\_02» за период с «2023-09-15 12:43:03» по «2024-05-11 12:43:23» см. таблица 1.

Таблица 1 – описательные статистики агрегата «WTG1» датчик «SENSOR\_02»

| **name** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| rms\_LOW\_SENSOR\_02 | 0,121362 | 0,077302 | 0,021654 | 0,045427 | 0,134274 | 0,185967 | 0,464757 |
| so1\_lss\_LOW\_SENSOR\_02 | 0,000000 | 0,000000 | 0,000000 | 0,000000 | 0,000000 | 0,000000 | 0,000000 |
| **so1\_iss\_LOW\_SENSOR\_02** | **0,010443** | **0,008829** | **0,000459** | **0,002329** | **0,011734** | **0,016740** | **0,093797** |
| so1\_hs\_is\_LOW\_SENSOR\_02 | 0,003038 | 0,002237 | 0,000919 | 0,002436 | 0,002880 | 0,003320 | 0,042933 |
| so1\_hss\_LOW\_SENSOR\_02 | 0,011321 | 0,001734 | 0,008255 | 0,010452 | 0,011256 | 0,011994 | 0,036929 |
| so2\_lss\_LOW\_SENSOR\_02 | 0,004433 | 0,003544 | 0,000166 | 0,002360 | 0,003756 | 0,005682 | 0,046956 |
| so2\_iss\_LOW\_SENSOR\_02 | 0,003390 | 0,004842 | 0,000388 | 0,001846 | 0,002723 | 0,003901 | 0,088050 |
| so2\_hs\_is\_LOW\_SENSOR\_02 | 0,006323 | 0,002299 | 0,001883 | 0,005236 | 0,006221 | 0,007274 | 0,037871 |
| so2\_hss\_LOW\_SENSOR\_02 | 0,004343 | 0,002362 | 0,002013 | 0,003496 | 0,004130 | 0,004632 | 0,039386 |
| so3\_lss\_LOW\_SENSOR\_02 | 0,014265 | 0,008249 | 0,001152 | 0,008559 | 0,013023 | 0,018303 | 0,087825 |
| so3\_iss\_LOW\_SENSOR\_02 | 0,007027 | 0,005523 | 0,000975 | 0,002660 | 0,007280 | 0,010450 | 0,075773 |
| so3\_hs\_is\_LOW\_SENSOR\_02 | 0,009895 | 0,001731 | 0,004875 | 0,008881 | 0,009861 | 0,010822 | 0,027810 |
| so3\_hss\_LOW\_SENSOR\_02 | 0,006750 | 0,001544 | 0,004145 | 0,006292 | 0,006655 | 0,007061 | 0,028526 |
| fband1\_LOW\_SENSOR\_02 | 0,115275 | 0,080814 | 0,013275 | 0,034610 | 0,131541 | 0,183769 | 0,426877 |
| fband2\_LOW\_SENSOR\_02 | 0,020004 | 0,007021 | 0,005369 | 0,016869 | 0,019821 | 0,022927 | 0,120657 |
| **fband3\_LOW\_SENSOR\_02** | **0,016178** | **0,003552** | **0,012053** | **0,015158** | **0,015888** | **0,016743** | **0,076161** |
| fband4\_LOW\_SENSOR\_02 | 0,003429 | 0,003645 | 0,002115 | 0,002769 | 0,002971 | 0,003254 | 0,061082 |
| fband5\_LOW\_SENSOR\_02 | 0,005925 | 0,003503 | 0,003736 | 0,005014 | 0,005469 | 0,006036 | 0,059869 |
| **peak2peak\_LOW\_SENSOR\_02** | **0,593829** | **0,485326** | **0,146575** | **0,312374** | **0,592309** | **0,811571** | **8,210626** |

Во всех столбцах, кроме «so1\_lss\_LOW\_SENSOR\_02», наблюдаются выбросы в данных, это можно увидеть также и на гистограммах. Выбраны 3 столбца для визуализации и подтверждения вывода о выбросах:

- «so1\_iss\_LOW\_SENSOR\_02» (см. рис. 2.1)

- «fband3\_LOW\_SENSOR\_02» (см. рис. 2.2)

- «peak2peak\_LOW\_SENSOR\_02» (см. рис. 2.3)

При сравнении гистограмм столбца «so1\_iss\_LOW» датчика «SENSOR\_02» агрегатов «WTG1» (рис. 2.1.) и «WTG2» (рис. 2.4) видно, что в первом случае имеются частотные повторения значений близких к нулю и часть значений, которые близки к 0.02, в то время, как у второго агрегата основная часть значений находится в около нулевых значениях. Исходя из этого можно сделать предположение, что в данных агрегата «WTG1» имеются аномалии.

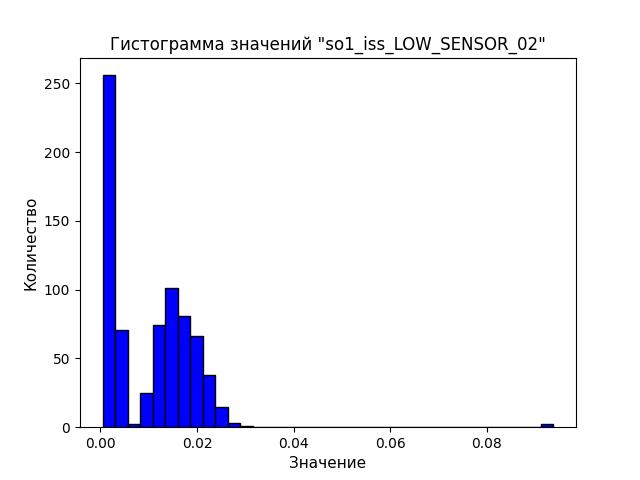


Рис. 2.1 – гистограмма столбца «so1\_iss\_LOW\_SENSOR\_02» агрегат «WTG1»

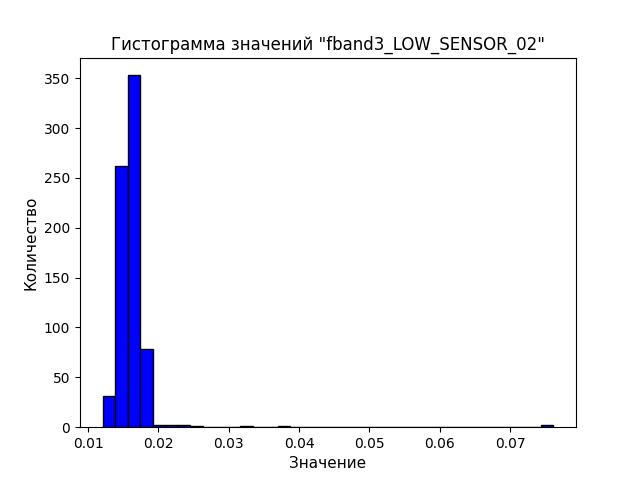


Рис. 2.2 - гистограмма столбца «fband3\_LOW\_SENSOR\_02» агрегат «WTG1»

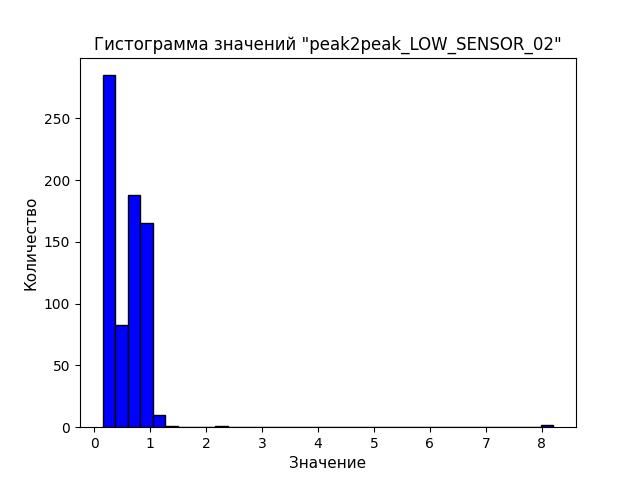


Рис. 2.3 - гистограмма столбца «peak2peak\_LOW\_SENSOR\_02» агрегат «WTG1»

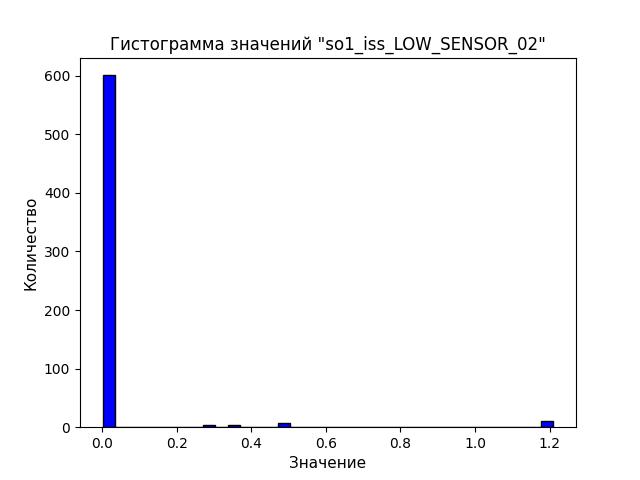


Рис. 2.4 – гистограмма столбца «so1\_iss\_LOW\_SENSOR\_02» агрегат «WTG2»

# 3. Описание обработки данных и разделение на обучающие и тестовые данные

Обработка данных:

Для обучения нейронной сети берутся низкочастотные данные (LOW) и для сравнения LOW и HIGH вместе, по датчику №2 (SENSOR\_02).

Берутся именно такие данные для того, чтобы можно было избежать нулевых значений при сравнении высокочастотных, фильтрованных и низкочастотных данных, в связи с этим не требуется принятия мер по удалению или замене нулевых значений.

За период с 15.09.2023 по 11.05.2024 гг. берутся данные следующих столбцов: rms, fband, peak2peak, so\_hs\_is, so\_hss, so\_iss, so\_lss. Данные столбцы являются наиболее значимыми из всего списка предоставленных параметров.

Для каждого агрегата и каждого изначального столбца создается список с условно-нормальными данными и на их основании рассчитываются следующие столбцы, для всех исходных столбцов, кроме tgnum:

1. Размах 80%;

2. Минимум;

3. Максимум;

4. Среднеквадратичное отклонение;

5. Изменение текущего значения по отношению к предыдущему – берутся исходные значения, а не условно-нормальные;

6. Предупреждение – присваивается на основании отношения текущего значения к предыдущему в сравнении с константой.

Разделение на обучающие и тестовые данные происходит в соотношении 80% первых значений на обучение, 20% последующих на тестирование. На обучение передаются данные со всех агрегатов, разделение идет итерационно, с последующей конкатенацией обучающих и тестовых данных. Это сделано для исключения перетасовки данных определенного агрегата.

Последовательное разделение сделано на основании того, что мы имеем дело с временными рядами.

После получения полного датасета с обучающими и тестовыми данными был применен метод фильтрации признаков «Information Gain» (IG) – который вычисляет уменьшение энтропии в результате преобразования данных.

Для дальнейшего тестирования было выбран один датасет с данными LOW и HIGH с IG больше 0.005 и три датасета только с данными LOW:

- полный набор данных;

- параметры, вычисленные IG больше нуля;

- параметры, вычисленные IG больше 0.005.

# 4. Описание решения и архитектура

Решение:

1. Создание дополнительного массива данных, со значениями условно-нормальных данных (определить математически, какие данные будут относиться к аномальным, а какие к условно-нормальным).
2. Вычисление метрик для дальнейшего обучения сети.
3. Разметка данных на нормальные и аномальные.
4. Разделение данных на тренировочные и тестовые.
5. Фильтрация признаков (IG).
6. Выбор типа нейросети.
7. Обучение нейросети.
8. Проверка точности: программно и визуально.
9. Внесение в таблицу сравнения параметров модели и точность.

Архитектура решения представлена на рисунке 4.1.

Блок-схема алгоритма представлена на рисунке 4.2-4.3.

Выбор модели нейросети был между базовой моделью ECOD библиотеки PYOD, моделью на основе SimpleRNN и моделью на основе LSTM с использованием полносвязных слоев. После проведения ряда тестов была выбрана модель SimpleRNN, которая более точно предсказывает значения аномалий после обучения и проверки визуальной и методом predict на тестовой выборке.

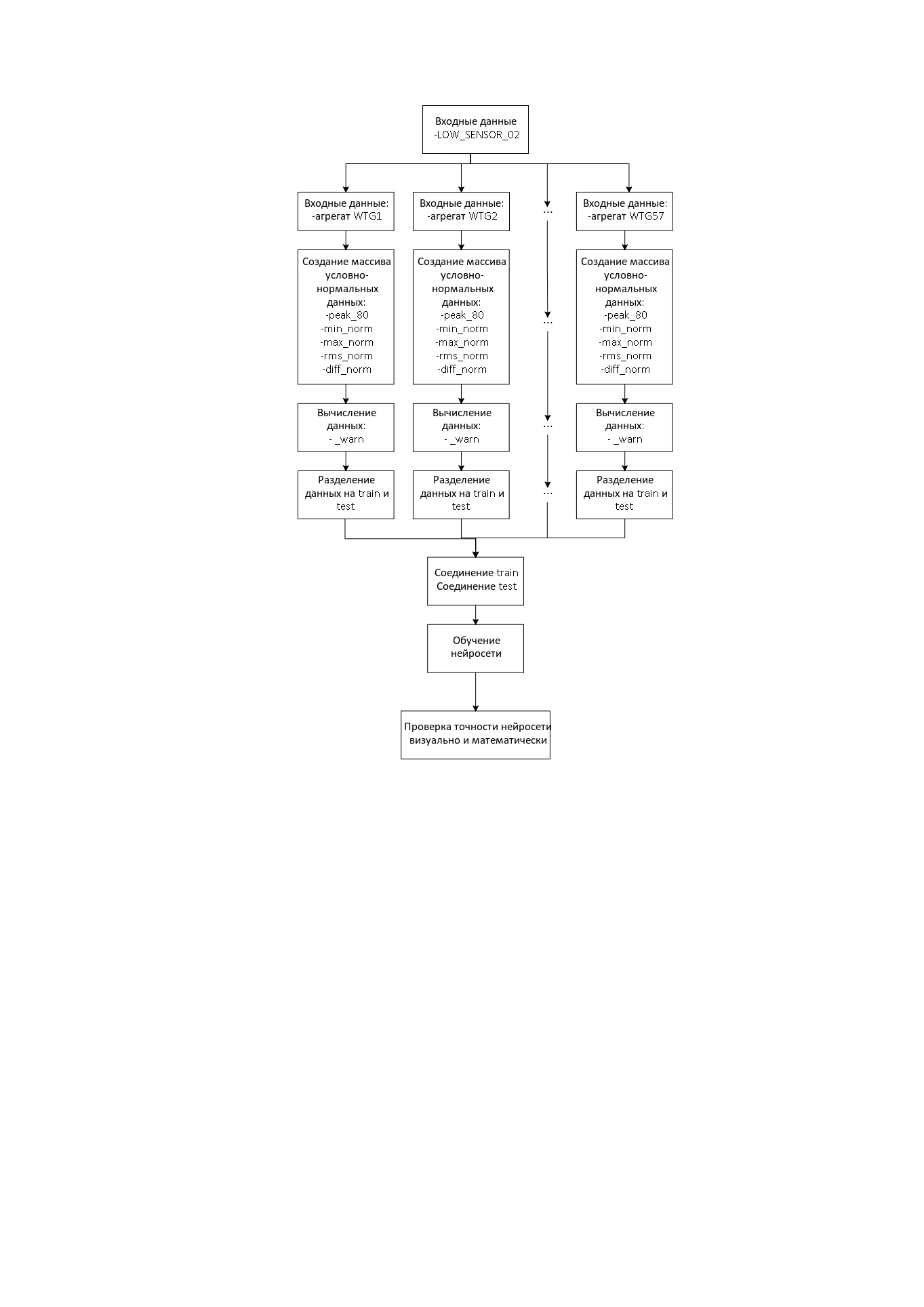


рис. 4.1 – архитектура решения

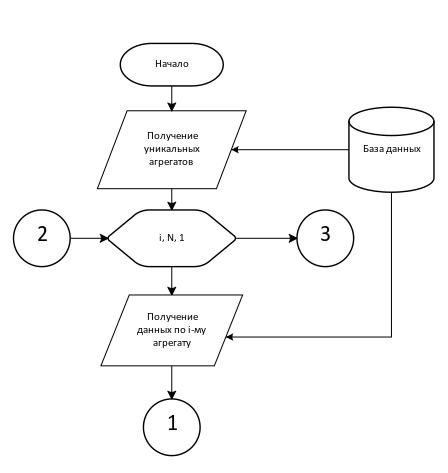


рис. 4.2 – блок-схема часть 1

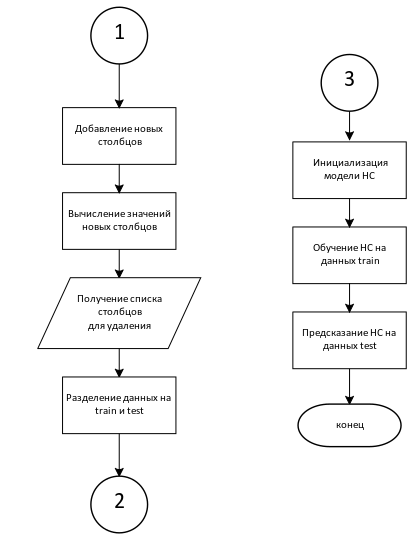


рис. 4.3 – блок-схема часть 2

# 5. Описание обучения

Данные для обучения берутся по всем агрегатам в низкочастотном диапазоне.

Описание процесса подготовки данных.

В цикле по количеству агрегатов выполняется следующий алгоритм:

1. По выбранному агрегату берутся данные столбцов: rms, fband, peak2peak, so\_hs\_is, so\_hss, so\_iss, so\_lss датчика «SENSOR\_02» данных «LOW» и «HIGH».

2. На каждый исходный столбец:

2.1. Инициализируется датафрейм условно-нормальных значений (при каждом переходе цикла);

2.1.1. Значение приравнивается к условно-нормальному, если абсолютное отношение текущего значения исходного столбца к значению предыдущего меньше константы (константа равна двум) и переменная «шаг» приравнивается к единице, в ином случае значение переменной «шаг» увеличивается на один.

2.2. Создаются столбцы для записи пяти вычисляемых значений (при первом вхождении цикла, начиная со второго, данные зануляются в текущих столбцах).

3. Считается изменение текущего значения по отношению к предыдущему (на основании исходных значений);

4. По каждому столбцу вычисляются метрики на основании значений из инициализированного в пункте 2.1. датафрейма:

4.1. Размах 80% - разница между квантилем 90% и квантилем 10% условно-нормальных значений.

4.2. Минимум – минимум условно-нормальных значений;

4.3. Максимум – максимум условно-нормальных значений;

4.4. Среднеквадратичное отклонение (СКО) – условно-нормальных значений;

4.5. Предупреждение – значение равно 1, если переменная «шаг» (из пункта 2.1.1) больше трех (для отсечения одиночных и парных выбросов) и 0 при значении переменной «шаг» равной единице.

5. Разделение получившегося фрейма данных на тренировочные и тестовые в соотношении: первые 80 % тренировочные, последующие 20% тестовые.

6. При первом вхождении цикла инициализируются переменные для хранения общих тренировочных и тестовых данных. Начиная со второй итерации, к инициализированным данным добавляются новые вычисленные данные.

7. Фильтрация признаков.

После завершения цикла по агрегатам, инициализируются модели обучения:

**1. Модель ECOD библиотеки Pyod**

Библиотека Pyod включает в себя более 40 алгоритмов обнаружения выбросов от классических LOF, PCA и kNN до новейших ROD, SUOD и ECOD.

ECOD – это непараметрический, легко интерпретируемый алгоритм обнаружения выбросов основанный на эмпирических функциях CDF, представленный в 2022 году.

Параметры данной модели:

- сontamination- cтепень искажения набора данных, т.е. доля отклонений в наборе данных. Используется при подборе для определения порогового значения для функции принятия решения.

- n\_jobs – кол-во заданий на параллельную работу.

**2. Модель на основе SimpleRNN библиотеки Keras**

SimpleRNN – Простаярекуррентная нейронная сеть, в которой выход предыдущего временного шага должен быть передан в следующий шаг (см. рис. 5.1).

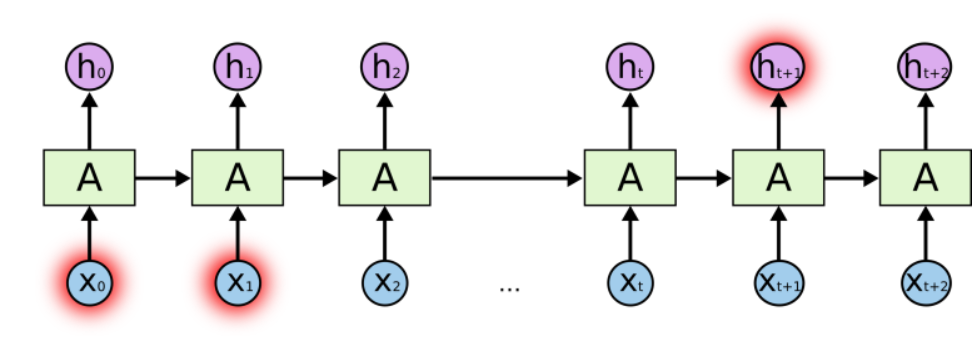


рис. 5.1 – архитектура модели SimpleRNN

Модель имеет один слой SimpleRNN и четыре полносвязных слоя Dense разной величины (8, 64, 256, 1).

На вход подаются данные размером (1, 109), где 109 – максимальное количество столбцов, задействованных на обучение. Используется 3 варианта данных: один без фильтрации, два после применения фильтрации признаков.

Последний полносвязный слой имеет размер равный 1, так как мы предсказываем нормальные и аномальные данные.

Модель принимает следующие параметрами:

batch\_size – размер блока данных;

validation\_split – размер валидационной выборки;

monitoring – параметр, который модель улучшает во время обучения;

epochs – количество эпох на обучение, стоит ограничение, если параметр monitoring не улучшился в течение 100 эпох, происходит завершение обучения.

**3. Модель на основе LSTM библиотеки Keras**

Long Short Term Memory (LSTM) - расшифровывается как «Долговременная кратковременная память». Теоретически это более «сложная» рекуррентная нейронная сеть, вместо простого повторения, в нем также есть «ворота», которые регулируют поток информации через модуль, как показано на рисунке 5.2.

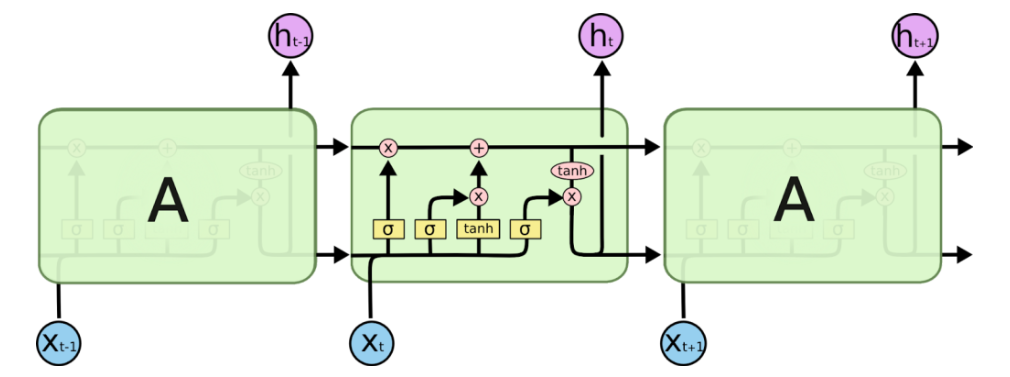


рис. 5.2 – архитектура модели LSTM

Модель имеет 2 слоя LSTM (64, 256) и два полносвязных слоя Dense (64, 1).

Входные данные и параметры аналогичны модели на основе SimpleRNN.

**Для моделей SimpleRNN и LSTM используются следующие функции потерь:**

1. Среднеквадратичная ошибка (MSE), вычисляемая по формуле:

где 𝑛 - количество наблюдений по которым строится модель и количество прогнозов,

– фактические значение зависимой переменной для 𝑖-го наблюдения,

 – значение зависимой переменной, предсказанное моделью.

2. Среднеквадратичная логарифмическая ошибка (MSLE), вычисляемая по формуле:

где 𝑛 - количество наблюдений по которым строится модель и количество прогнозов,

– фактические значение зависимой переменной для 𝑖-го наблюдения,

  — значение зависимой переменной, предсказанное моделью.

3. Бинарная перекрестная энтропия (BCE), вычисляемая по формуле:

где 𝑛 - количество наблюдений по которым строится модель и количество прогнозов,

– фактические значение зависимой переменной для 𝑖-го наблюдения,

- прогнозируемая вероятность того, что i-е наблюдение будет относиться к классу 1.

Все результаты обучения добавляются в датафрейм, с указанием использованных параметров и итоговых метрик точности.

Было опробовано более 10 вариаций данных моделей с различными параметрами и функциями потерь.

# 6. Описание итогового результата

После обучения, происходит прогнозирование данных и подсчет точности предсказания. Этапы работы:

Первый этап – прогнозирование на тренировочных данных.

Второй этап – прогнозирование на тестовых данных.

Третий этап – визуализация прогноза по всем агрегатам.

Результаты работы использованных моделей с различными параметрами представлены в таблице 6.1.

Таблица 6.1 – результаты работы моделей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Модель** | **loss** | **optimizer** | **train** | **test** | **Размер** |
| 13 | **LSTM, model13: sensors=[low], X: all** | **MSE** | **rmsprop** | **0,9833** | **0,9716** | **8** |
| 3 | SimpleRNN, model3: sensors=[low] X: all | MSE | rmsprop | 0,9827 | 0,9713 | 8 |
| 4 | SimpleRNN, model4: sensors=[low], X: IG>0 | MSE | rmsprop | 0,9825 | 0,9712 | 8 |
| 8 | SimpleRNN, model8: sensors=[low], X: all | MSE | rmsprop | 0,9824 | 0,9712 | 16 |
| 5 | SimpleRNN, model5: sensors=[low], X: IG>0.005 | MSE | rmsprop | 0,9835 | 0,9710 | 8 |
| 7 | SimpleRNN, model7: sensors=[low], X: IG>0.005 | MSE | adam | 0,9831 | 0,9674 | 8 |
| 6 | SimpleRNN, model6: sensors=[low], X: all | MSE | adam | 0,9827 | 0,9670 | 8 |
| 14 | LSTM, model14: sensors=[low], X: all | MSE | adam | 0,9823 | 0,9669 | 16 |
| 10 | SimpleRNN, model10: sensors=[low], X: all | BCE | rmsprop | 0,9799 | 0,9667 | 8 |
| 11 | SimpleRNN, model11: sensors=[low], X: all | BCE | adam | 0,9825 | 0,9667 | 8 |
| 12 | LSTM, model12: sensors=[low], X: all | MSE | adam | 0,9832 | 0,9667 | 8 |
| 9 | SimpleRNN, model9: sensors=[low], X: all | MSLE | rmsprop | 0,9631 | 0,9300 | 8 |
| 1 | PYOD.ECOD, model2 | - | - | 1,0000 | 0,9297 | 1 |
| 2 | SimpleRNN, model3\_01: sensors=[low, high], X: IG>0.005 | MSE | rmsprop | 0,9465 | 0,9204 | 8 |
| 0 | PYOD.ECOD, model | - | - | 1,0000 | 0,8586 | 1 |

Из таблицы можно увидеть, что наилучшей моделью из отработанных оказалась модель на основе LSTM, которая обучалась на данных sensor LOW. Точность на тестовых данных составила приблизительно **0,9716**. График предсказанных значений представлен на рисунке 6.2. Из него можно заключить, что модель, хорошо предсказывает продолжительные аномалии, но плохо справляется с непродолжительными аномалиями, которые возможно являются единичными выбросами.

Применения на практике данное решение не имеет, модель требует дополнительной настройки для более точного предсказания.

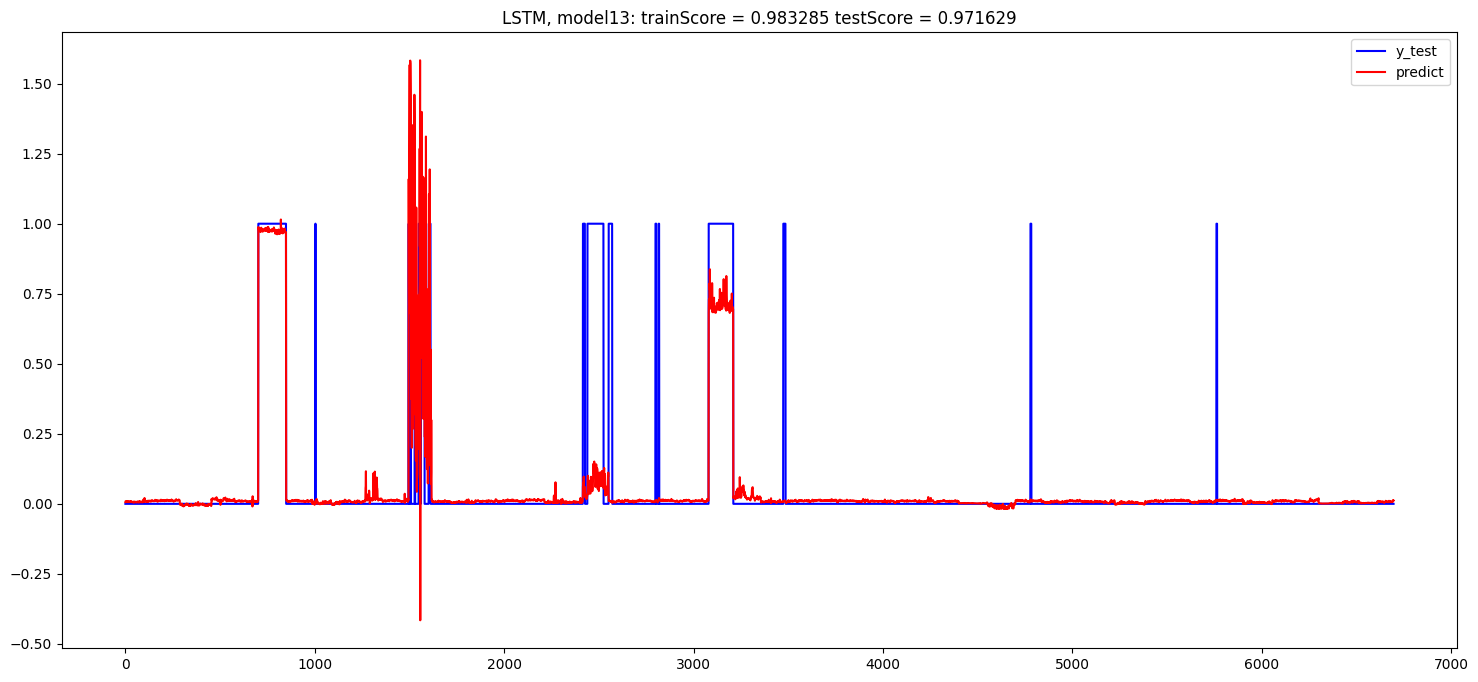


Рис. 6.2 – результат предсказания нейросети LSTM, model13

# 7. Заключение с выводами и планами на дальнейшее развитие

В целом модели показывают хорошие показатели по точности предсказания, в среднем уровень точности равен 0.96-0.97, но на некоторых агрегатах мало исходных данных или требуется другой метод для выявления аномалий.

Дальнейшее развитие вижу по следующим направлениям:

1. Изменение метода выявления аномалий, что повысит точность предсказания.

2. Сделать индивидуальный подход, с вычислениями для каждого агрегата, что позволит более полно раскрыть имеющиеся аномалии.

3. После реализации пункта 1 или 2 (что будет давать более точные предсказания) сделать вариативный детектор, который не просто будет показывать аномалию, а и предсказывать какая именно неисправность могла дать такой результат.

# 8. Источники, использованные при разработке

1. А.В. Барков, НА. Баркова, А.Ю. Азовцев. Мониторинг и диагностика роторных машин по вибрации, 2023. – 160 с.
2. Питер Брюс, Эндрю Брюс. Практическая статистика для специалистов Data Scienсe [Practical Statistics for Data Scientists], 2020. – 304 c.

**Internet – ресурсы**

1. Сайт о программировании [Электронный ресурс] URL: <https://www.python.org/>
2. Сайт о программировании [Электронный ресурс] URL: <https://keras.io/api/layers/recurrent_layers/simple_rnn/>
3. Сайт о программировании [Электронный ресурс] URL: <https://www.datatechnotes.com/2018/12/rnn-example-with-keras-simplernn-in.html>
4. Сайт о программировании [Электронный ресурс] URL: <https://adtk.readthedocs.io/en/stable/notebooks/demo.html>
5. Сайт о программировании [Электронный ресурс] URL: <https://adtk.readthedocs.io/en/stable/>
6. Сайт о программировании [Электронный ресурс] URL: <https://keras.io/api/losses/probabilistic_losses/#binary_crossentropy-function>
7. Сайт о программировании [Электронный ресурс] URL: <https://habr.com/ru/companies/otus/articles/570314/>
8. Сайт о программировании [Электронный ресурс] URL: <https://loginom.ru/blog/quality-metrics>
9. Сайт о программировании [Электронный ресурс] URL: <https://www.geeksforgeeks.org/binary-cross-entropy-log-loss-for-binary-classification/>
10. Сайт о программировании [Электронный ресурс] URL: <https://proglib.io/p/postroenie-i-otbor-priznakov-chast-2-feature-selection-2021-09-25>
11. Сайт о программировании [Электронный ресурс] URL: <https://pyod.readthedocs.io/en/latest/pyod.models.html>
12. Сайт о программировании [Электронный ресурс] URL: <https://habr.com/ru/articles/487808/>