**Поиск аномалий при добыче**

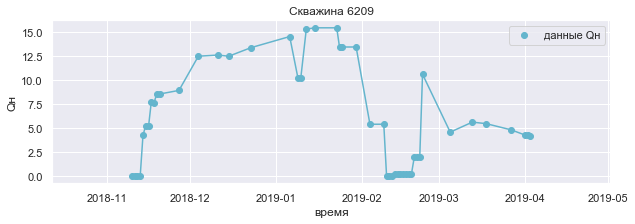
**Определение данных для работы**

Идентификация аномалий предполагает работу с данными. Предоставим краткое описание данных, используемых в поставленной задаче. Для конкретизации задачи рассмотрим месторождения, оборудованные УЭЦН (погружная насосная установка). Основными признаками, описывающими работу на конкретной скважине, являются:

* Месторождение – название месторождения;
* Скважина – название скважины;
* Дата – дата проведенного измерения;
* Qн – дебит нефти;
* Qж – дебит жидкости;
* I – сила тока погружного электродвигателя насосного агрегата;
* U – напряжение погружного электродвигателя насосного агрегата;
* Активная мощность;
* Частота вращения;
* Давление на приеме насоса
* Коэффициент загрузки двигателя насоса (отношение всей мощности, затрачиваемой на холостую и полезную работу оборудования, к суммарной установленной мощности двигателей);
* Активную мощность двигателя (часть полной мощности за период времени, определяемый частотой тока двигателя насоса);

Все эти данные могут быть искажены вылетами (выбросами), пропуском измерений из-за ошибок телеметрии и прочими нерегулярными искажениями, которые сигнализируют о нарушении работы погружного насоса, как следствие – добычи нефти.

Прежде всего нас интересует дебит нефти в заданный момент времени на конкретном месторождении в конкретной скважине. Остальные признаки являются вспомогательными в исследовании.



**Рисунок** 1 - Пример графика дебита нефти

Заметим, что наличие даты является ключевой особенностью данных. Так как процесс добычи нефти протяжен по времени, то значения признаков, таких как Qн, Qж в данный момент времени зависят от значений в предыдущие моменты времени. Такого рода данные называют временными рядами (Time Series). Также важно, что в задаче требуется определять аномалии в момент поступления данных, а значит для этого можно использовать лишь данные прошедших измерений. Методы, обладающие таким свойством, называют онлайн-методами.

**Алгоритмы поиска аномалий**

Напомним, что первоочередной задачей исследования является задача поиска аномалий в данных с пропусками и нерегулярными измерениями.

Рассмотрим примеры алгоритмов для поиска аномальных данных и возможность их применения в данной работе.

1. LocalOutlierFactor – Популярный алггоритм на основе метода k-ближайших соседей. Выбросом является значение с наибольшим расстоянием от максимальных плотностей данных;
2. IsolationForest – Метод поиска выбросов путем их изоляции решающим деревом;
3. Правило 3-х сигма – статистически известно, что в диапазоне [M - 3σ; M + 3σ] содержится более 99% данных, тогда как остальные данные вне этого диапазона – аномальные;
4. Межквартильный размах (IQR = Q3 – Q1) – в статистике применяется для поиска данных не отвечающих нормальному распределению. Выбросы – данные вне 1.5\*IQR от первого и третьего квартиля (Q1, Q3 соответственно);
5. Нейронные сети – о них отдельно;

Так как мы рассматриваем онлайн данные, то логично использование приведенных выше функций в виде оконных функций вычисления. Параметр длинны окна необходимо определить на исторических данных и понять его пригодность для промышленного использования.

Для более достоверной детекции аномальных данных предлагается использование двух разных окон: малой и большой длины. Например, 15 и 60 минут, что позволит определить аномалии в данных на средне- и краткосрочном графике и избежать пропуска долгосрочного накопления ошибки (например, ступенька в данных).

**Алгоритмы поиска аномалий – нейронные сети**

Задача использования нейронных сетей в поиске аномалий достаточно непростая в основном из-за особенностей получаемых данных. В этом случае большую роль играет препроцессинг, о нем и пойдет речь. Выбор модели для идентификации аномалии здесь не рассматриваю.

Нейронной сети нельзя передать вектор с отсутствующими значениями, так как иначе совершенно непонятно, как будет происходить обучение. Из этого можно сделать вывод, что нужно как-то преобразовывать данные. Для каждого введем вектор такой, что

Таким образом в модель вносится информация о том, где значения пропущены, а где нет. При этом реальные пропуски в данных заменяются нулем. Это корректное преобразование, так как само значение ноль использоваться моделью не будет.

В силу того, что работа производится с данными, в которых встречаются аномалии, а пропусков довольно много, то стоит задуматься об аугментация данных, чтобы расширить выборку. Для решения данной проблемы введем новый случайный вектор , который получается в результате аугментации – добавления искусственных аномалий к вектору . Аномалии можно создавать путем внесения случайной логнормальной ошибки. Логнормальное распределение предлагается, так как возможны сильные выбросы в данных, которые стоит отразить при расширении выборки.

Так как рассматриваемые признаки имеют разные величины измерений, то требуется задуматься об нормализации данных. Поскольку в данных присутствуют выбросы, то традиционные способы, такие как:

* StandardScaler, который заключается в вычитании среднего и делении на стандартное отклонение;
* MinMaxScaler который заключается в вычитании минимума и деление на диапазон значений признака (разность максимального и минимального значений)

не работают, поскольку эти методы сильно подстраиваются под присутствующие в данных выбросы.

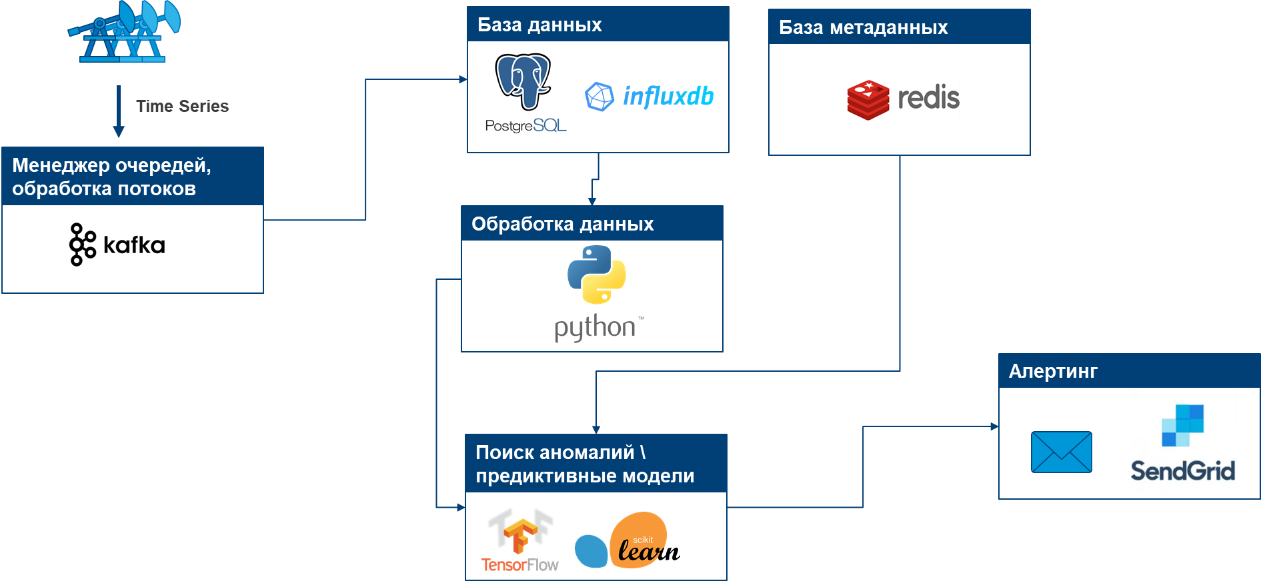
Предлагается использовать робастный метод RobustScaler, который заключается в исключении по % минимальных и максимальных значений перед методом StandartScaler. Такой способ скалирования гарантированно не подстраивается под выбросы, если в данных присутствует не более выбросов.

Что касается выбора какой-то одной конкретной функции оценки выбросов, то я бы не ограничивался единым параметром мониторинга и поиска выбросов. У всех методов различная степень чувствительности к выбросам и устойчивость их определения.

Предлагается делать вывод о выбросе в данных по наличию детекции минимум в 2-х предупреждающих сигналах.

**Архитектурное решение**

Предлагается следующий вариант архитектурной диаграммы ():

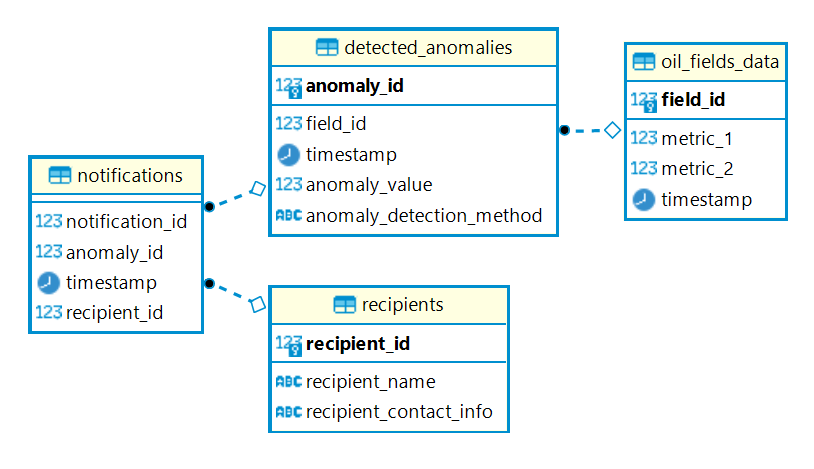


**Рисунок** 2 – Верхнеуровневая архитектура решения

В этой архитектуре данные потоковой добычи собираются и проходят анализ временных рядов после обработки для выявления аномалий. В случае обнаружения аномалий создаются оповещения, создается соответствующая запись в БД и данные о аномалии отправляются ответственным лицам.

**Модель данных**

Модель данных включает четыре основные таблицы: таблицу для хранения потоковых данных месторождений, таблицу для обнаруженных аномалий, таблицу для уведомлений и контакты лиц для уведомления. Вот модель данных в виде диаграммы ():



**Рисунок** 3 – Модель данных PostgreSQL

В этой модели, "Oil Fields Data" содержит информацию о показателях по добыче для каждого месторождения, "Detected Anomalies" сохраняет информацию об обнаруженных аномалиях, а "Notifications" используется для отслеживания уведомлений, отправленных ответственным лицам. "Recipients" хранит данные о лицах, которым следует отправлять уведомления. Поля “Metric\_<0…N>” отвечают отслеживаемым параметрам.

Когда система обнаруживает аномалии, она сохраняет соответствующую информацию в таблице "Detected Anomalies" и отправляет уведомления с подробностями аномалии, используя информацию из "Notifications" и "Recipients". Система уведомлений может быть построена с использованием сервисов электронной почты или мессенджеров.

Возможно, что в будущем стоит рассмотреть применение БД для хранения временных рядов, например, InfluxDB предоставляет эффективное решение для хранения временных рядов, таких как потоковые данные с месторождений. Но я с ней не знаком ☹

Для решения этой задачи можно использовать следующие технические компоненты:

1. База данных: Для хранения проведенных обнаруженных аномалий и пороговых уровней можно использовать реляционные базы данных, такие как PostgreSQL, из-за их надежности и производительности.
2. Хранение конфигурационных данных настроек алгоритмов (в том числе хранения пороговых уровней) можно использовать базу данных, такую как Redis, для быстрого доступа к ключевым значениям. Это позволит эффективно хранить и обновлять пороговые уровни для каждого месторождения независимо.
3. Apache Kafka: Для обработки потоковых данных и управления потоком данных из разных месторождений можно использовать Apache Kafka. Он поможет обеспечить надежную и масштабируемую платформу для обработки потоков данных и обеспечит устойчивую систему, способную справиться с большими объемами данных.
4. Python + ML Framework (pytorch, TF) + sklearn: Для анализа временных рядов и выявления аномалий можно использовать библиотеки Python.
5. Система уведомлений, такая как SendGrid или рассылка e-mail через корпоративный почтовый сервис для отправки уведомлений ответственным лицам
6. Возможно, потребуется интеграция с корпоративной системой, например, ACTIVE Directory