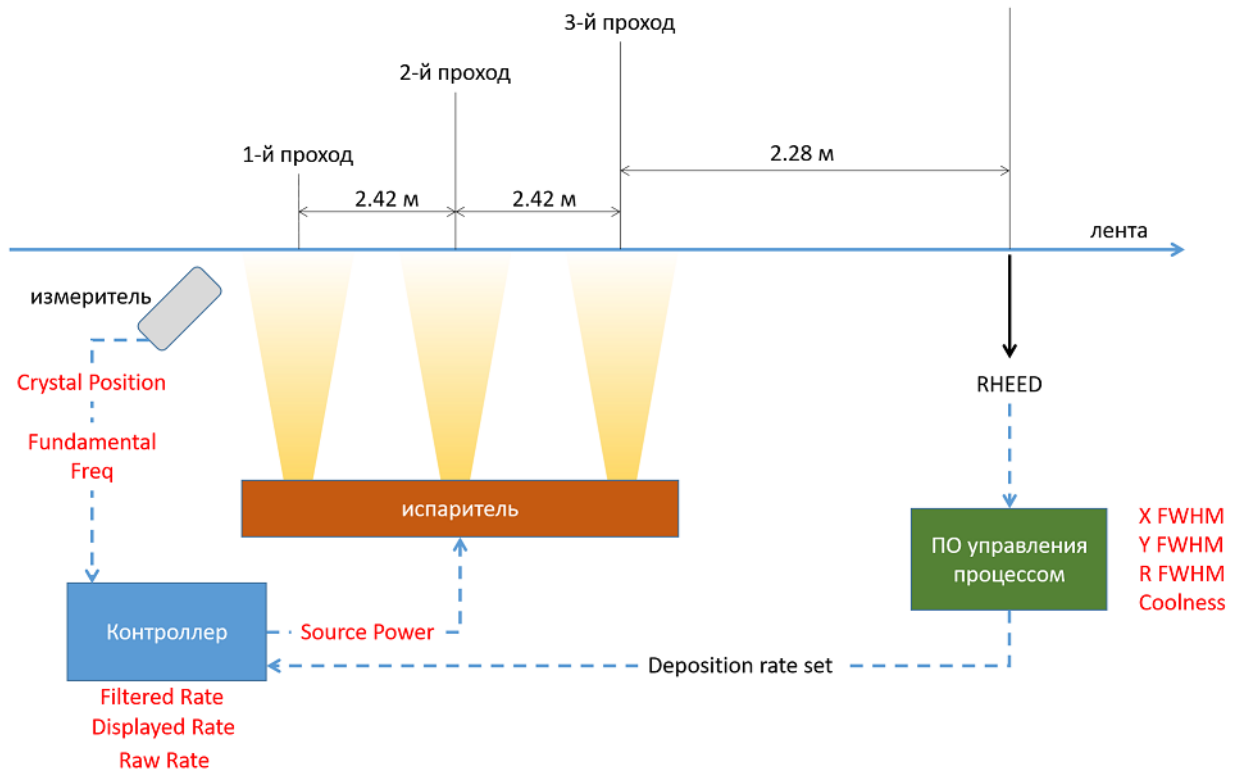


В рамках задачи рассматривается [вакуумное напыление](#) с ионным ассистированием, процесс IBAD (Ion Beam Assisted Deposition). Суть процесса заключается в том, что лента едет по зоне осаждения и испаритель осаждает на ленту оксид магния. Лента едет по так называемой улитке. Т.е. проехав зону осаждения один раз, лента разворачивается и проезжает эту же зону осаждения второй, 3-й разы. Конкретно для изучаемого процесса лента проезжает зону 3 раза. На отсутствуют развороты, но важно понять суть, что события в испарителе одновременно действуют на три участка ленты, которые удалены друг от друга на 2.42 м.



Испаритель работает на мощности **Source Power** (все параметры, которые присутствуют в датасете выделены красным). Эту мощность задаёт контроллер. Контроллер получает данные по частоте резонатора (**Fundamental Freq**) от измерителя и по падению частоты определяет скорость осаждения. Измеритель имеет расходный элемент резонатора, который периодически переключается оператором, что записывается в параметр **Crystal Position**. По мере работы резонатора стабильность его частоты ухудшается и измерения становятся шумными, поэтому оператор переключает его на следующий. Контроллер отдаёт сразу три измеренных скорости: **Filtered Rate**, **Displayed Rate** и **Raw Rate**. У контроллера есть постоянная времени фильтра скорости осаждения. Она зафиксирована для всех процессов. На основе измеренной скорости осаждения и уставки по скорости осаждения контроллер рассчитывает мощность испарителя (**Source Power**). Чем больше мощность, тем больше скорость осаждения.

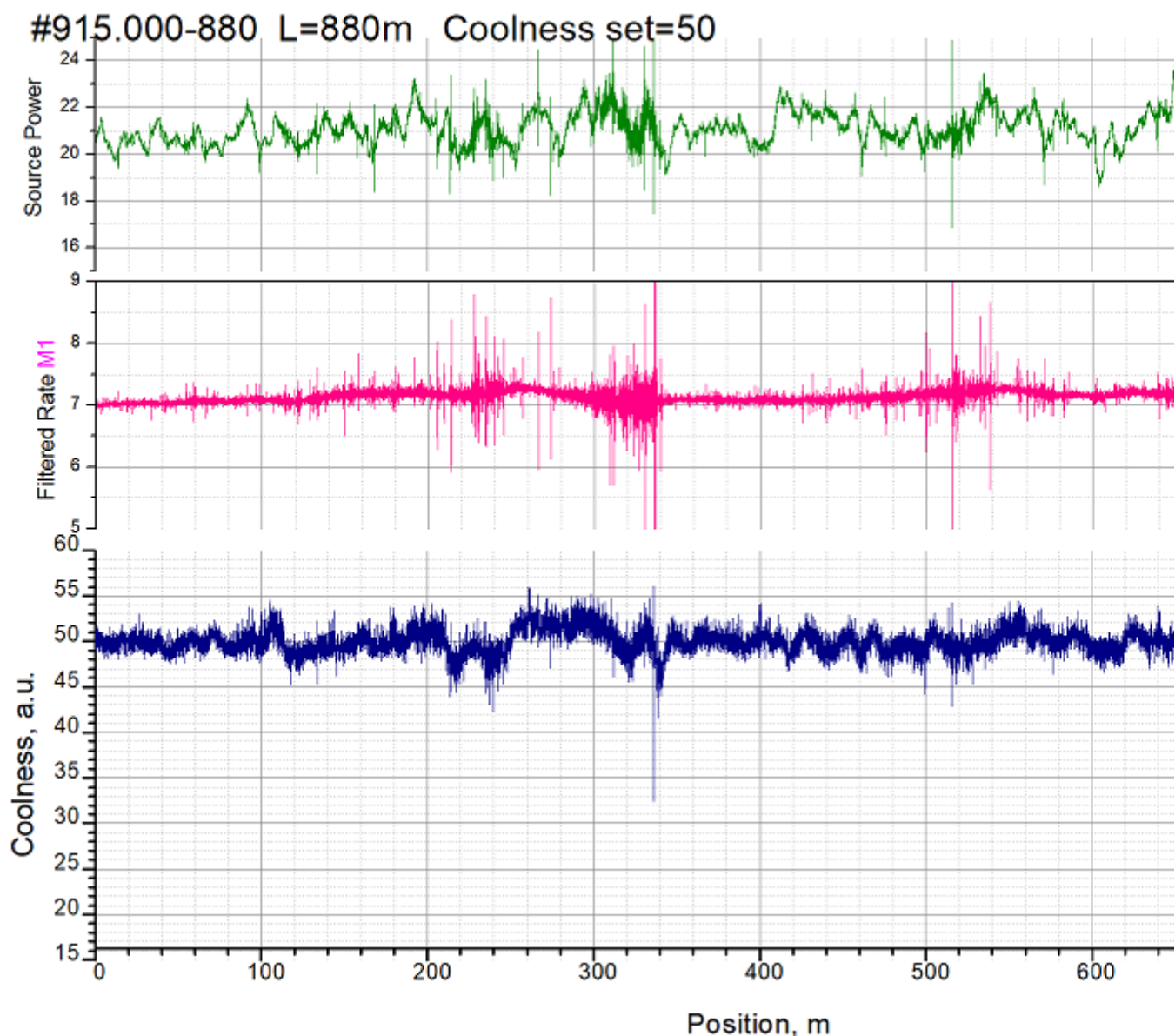
На самом деле в эту же зону осаждения светит ионный источник. Этот ионный источник травит слой, который осаждает испаритель. Т.е. отсутствие или снижение осаждения, не означает, что просто ничего не происходит. В этом случае происходит травление слоя полученного на предыдущих проходах или даже предыдущих процессах. Кроме того, травления происходит неравномерно по проходам зоны осаждения. 1-й и 3-й проходы имеют типично около 70% скорости травления от 2-го прохода.

Под таким процессом лента проезжает три раза и через 2.28 м от 3-го прохода качество полученного слоя анализируется дифракционным инструментом RHEED ([Reflected High Energy Electron Diffraction](#)).

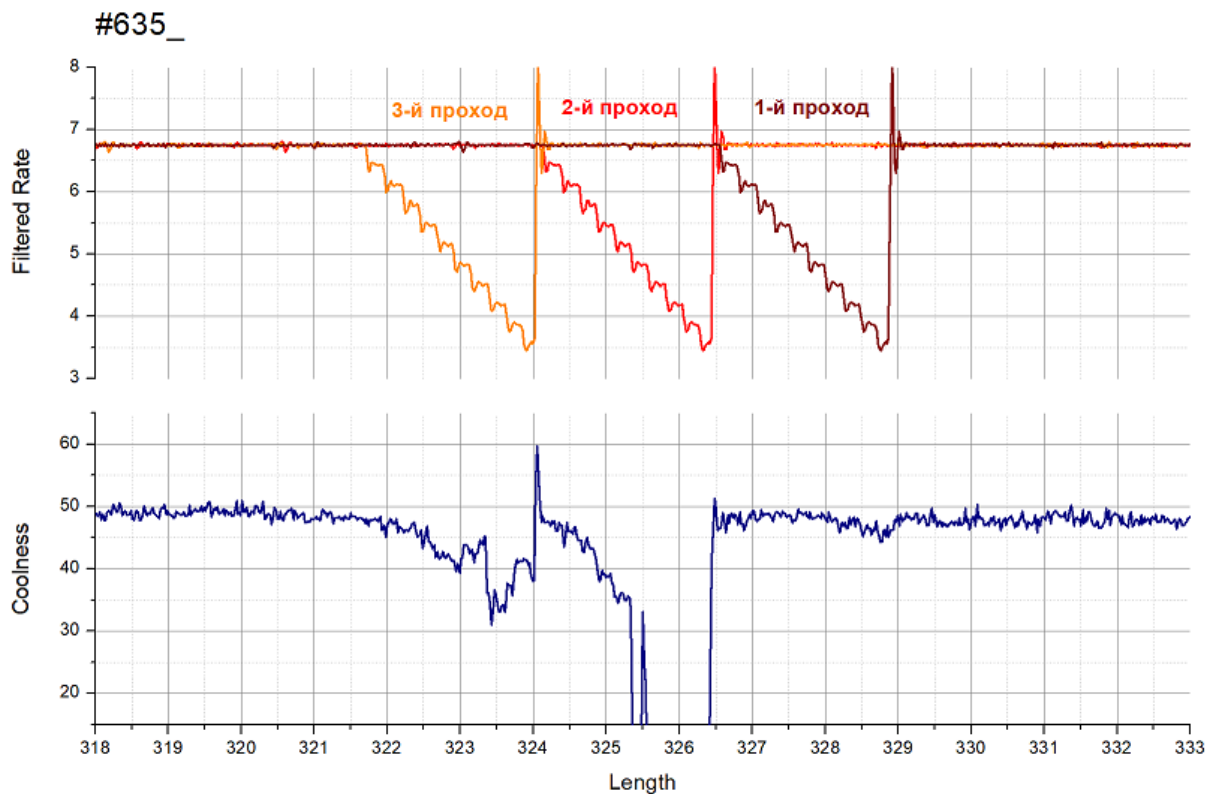
Софт управления измеряет три сечения выбранного пика (**X FWHM**, **Y FWHM** и **R FWHM**). FWHM - Полная ширина на уровне половинной амплитуды (англ. FWHM — full width at half maximum). На основе пика также рассчитывается параметр **Coolness** («холодок»). Софт управления имеет уставку по параметру Coolness и меняет уставку скорости осаждения в контроллере. Параметры дифракции в RHEED определяются кооперативным влиянием процессов на всех проходах. В общем случае, чем больше скорость осаждения, тем больше Coolness (на самом деле есть максимум на высоких скоростях осаждения, но по факту не применяется). Характеристика Coolness(Rate) не линейная, чем меньше скорость осаждения и Coolness тем круче зависимость, а работать охота именно на низких значениях Coolness. Собственно прогноз Coolness является основной задачей. Типично наши логи строятся не от времени, а от координаты ленты (**Length**).

Во данных изначально присутствуют два типа логов (#915_M1 – лог испарителя и #915_RHEED – лог анализатора качества). Для задачи они уже сведены в один датасет, те колонки которые присутствовал в логге RHEED помечены как название колонки+_RHEED. Важно, в этих логах координата для событий испарителя сдвинута так, как будто они происходят на 2-м проходе ленты, а внутренне мы понимаем, что эти же события происходят и 2.42 м назад, и 2.42 м вперёд по координате. В принципе, для каждой строчки есть время записи этого параметра и оно остается истинным.

Получается такое распределение, например:



Для более полного понимания, ниже описан ещё специальный эксперимент искусственного возмущения. Был симитирован провал по скорости осаждения. Уставка скорости осаждения линейно снижалась примерно один период между проходами, затем возвращалась в регулярное значение. Для наглядности лог испарителя продублирован для всех проходов. Лента едет меньшей координатой вперёд, поэтому в анализатор RHEED сначала попадает часть ленты, которая воспринимала возмущение на 3-м проходе. Видно, что есть возмущение на Coolness, но текстура не исчезла, т.е. стравилась часть слоя. RHEED с 2-го прохода показывает, что слой был полностью стравлен и даже повреждён слой предыдущего процесса (места, где coolness не определён, т.е. = 0). Ну и наконец. Возмущение на 1-м проходе практически не повлияло на Coolness, т.к. слой успел нарости во 2-м и 3-м проходах.



Задача:

В рамках задачи необходимо прогнозировать уменьшение параметра холодок ("Coolness", среднее за 50 измерений, примерно 1 метр) на 5 более по сравнению со средним значением за последние 1 метра, через 435 измерений, примерно 9 метров относительно точки измерения.

Формат входа: два массива значений аналогичные входным данным. Формат выхода: вероятность относительно уменьшения значения более чем на 5 на 9 метров, относительно последней метки в массивах. Глубина данных для анализа, которые будут передаваться 10.000 замеров (примерно 23 метра).

Целевая метрика ROC AUC.

Решения необходимо присылать на платформу <https://sim.avt.global>

Поскольку подсчет баллов в финале отличается от подсчетов баллов в предыдущие этапы, то логин в платформу через платформу Талант для финальной задачи не поддерживается. Вам необходимо завести учетную запись (с любой почты), указав в имени название своей команды (точно совпадающие с заявленным) и отправлять решения с этого логина.

Для загрузки решения необходимо создать архив, содержащий файл `submission.py`, в котором представлено ваше решение. В этом файле должен быть объявлен класс `Predictor` с функциями `init(self)` и `forecast(self,df)`

`init(self)`, нужен для инициализации класса, вызывается один раз за тестирование, нужен чтобы загрузить все модели, для экономии времени.

`forecast(self,df)` нужен для генерации прогноза. В качестве `df` в функцию `forecast` передается pandas DataFrame, на 10000 значений, с колонками соответствующим данным. DataFrame полностью аналогичен приведенному, в нем могут быть пропуски, замены лент и т.д. В качестве возвращаемого значения функция `forecast` возвращает одно число от 0 до 1 - вероятность уменьшения более чем на 5 усредненного Coolness за ближайшие 9 метров.

С учетом большого количества вопросов про целевую переменную, поясняем дополнительно: Целевая переменная равна 1(True) если усредненное значение параметра Coolness за последние 50 измерений больше чем на 5 среднего значения

Coolness в 385-435 измерениях после данного момента в ином случае 0 (false). Иными словами для каждой строчки можно подсчитать среднее значение Coolness за последние 50 показателей и если среднее сейчас больше чем на 5 среднего через 435 строчек.

Вычисление этого параметра, можно представить так:

```
cool_col= rheed_long.rolling(50).mean()
```

```
y = ((cool_col- cool_col.shift(-435))>5), где rheed_long - входные данные.
```

Метрика по которым будут сравнивать решения - roc_auc (она измеряется от 0 до 1, чем больше, тем лучше).

Финальные баллы начисляются согласно регламенту, относительно лучшего решения среди участников.

Пример корректного submission, содержащий модель и класс, поддерживающего созданную модель прикладываем

(<https://drive.google.com/file/d/1QKBUYRGXTIXJ4HsFZ7T2EsEEOLFdw1o-/view?usp=sharing>). Для корректной работы модели необходимо хранить список колонок.

Прикладываем базовое решение

(https://drive.google.com/file/d/1q0BoGzJvVBjsSUS0x5IVL_d9vweNejB_/view?usp=sharing)