GANs for Trading Stratigies Backtesting

Ярослав Лисняк, Николай Аверьянов

26 марта 2022 г.

1 Постановка задачи и мотивация

Вообще говоря, временным рядом принято считать векторозначный случайный процесс

$$S_t = (S_t^1, \dots, S_t^n)$$

Любой датасет представляет собой некоторую реализацию этого случайного процесса, конкретный вероятностный исход. Имея подобные дискретные реализации мы бы хотели что-то понять про свойства вероятностного пространства и самого процесса S.

Важно понимать, что с подобной вероятностной постановкой будет некорректно решать задачу прогнозирования, сравнивая временные ряды как "длинные векторы например, считая расстояния между ними поточечно. Нам куда важнее обращать внимание на их вероятностные и прикладные характеристики: моменты, автоковариационные зависимости, стационарность, созависимости между координатами и многое другое. Правдоподобные траектории, построенные нашей моделью должны быть **следствием** хорошего выявления перечисленных выше характеристик, но ни в коем случае не основной целью модели.

Анализ временных рядов находит множество приложений, в частности в сфере финансов и финансовой математики. Так, мы будем генерировать систему временных рядов, стремясь получить последовательности, сохраняющие важные свойства. Если получится удачно аппроксиммировать распределение исследуемых рядов, то будет возможность иметь в распоряжении синтетические данные, достаточно качественные, для бэктеста торговых стратегий. В большинстве случаев, бэктестятся стратегии на исторических данных, то есть на одном вероятностном исходе. Расширить выборку синтетикой может оказаться крайне полезно.

2 Бейзлайн решение: что сделали авторы

Авторы используют 2 вида WGAN-ов (помимо RBF, которым они тоже пользуются в статье) с несложной структурой в виде полносвязных сетей и одномерных сверточных сетей.

Учатся они на небольшом двумерном датасете, состоящим из индексов за определенный промежуток времени. Они подают в свой WGAN в качестве условия некоторую последовательность предыдущих значений и выдают сэмплы на несколько шагов вперед. В статье они используют стратегию скользящего окна: взяв первые n сэмплов в качестве условия, они генерируют на k шагов вперед и дальше берут новые условные n сэмплов из сконкатенированных старых и сгенерированных (n с конца).

3 Идеи расширения

Несмотря на то, что размер статьи составляет 72 страницы, решение авторов не обладает сложной архитектурой и проверяется на данных небольшой размерности: модель принимает шум и 20 точек на вход, дает 5 точек на выход.

В связи с этим, решение имеет множество степеней свободы для расширения исследования. В этом разделе обсудим, в каком направлении мы пробовали двигаться.

3.a Архитектуры

Мы сконцентрировались на GAN-ах и WGAN-ах. **Для бейзлайна** взяли 2 архитектуры авторов: вариации MLP и сверточных сетей.

В качестве альтернативы нами была взята стандартная архитектура на основе трансформеров.

3.b Новые мета-признаки

Лосс и процесс понимания моделью того, как устроена наша система рядов для нас черный ящик и в целом, у модели есть множество способов генерировать траектории так, чтобы идти в сторону понижения лосса, далеко не все траектории при этом будут осмысленными. На этот счет авторы упоминают, что в случае с финансовыми временными рядами крайне важна разметка над данными и репликация моделью множества вероятностных характеристик рядов. В связи с этим, кажется разумным обратить внимание модели на отдельные статистики рядов. Так, мы решили добавить мета-признаки рядов в функцию ошибки некоторых из моделей и посмотреть, как это повлияет на качество обучения.

3.c Мета признаки, что мы опробовали:

Пусть $S_{\text{fake}}(t)$ – значение сгенерированного векторозначного ряда в момент времени t. $S_{\text{real}}(t)$ – значение реального векторозначного ряда в момент времени t. $\Sigma(S(t))$ – ковариационная матрица между координатами ряда $S(t)$. Зависимость от t можно убрать в случае стационарных данных. $Q_q(S)$ – вектор q-квантилей.

- **Моменты** ряда (пробовали в том числа абсолютные и центрированные). В лосс шла норма разницы векторов матожиданий фейковых и реальных рядов.

$$M = \int\limits_{t \in T} ||\mathbb{E}S_{\text{real}}(t) - \mathbb{E}S_{\text{fake}}(t)||dt$$

- **Ковариационные функции** между координатами ряда. В лосс шла матричная норма разницы ковариационных матриц.

$$C = \int\limits_{t \in T} ||\Sigma(S_{\text{real}}(t)) - \Sigma(S_{\text{fake}}(t))||dt$$

- **Авторегрессионные функции**. Каждому ряду сопоставлялись авторегрессионные операторы, в лосс шла норма разницы операторов, построенных по фейковым и реальным данным.

$$A(\tau) = \int\limits_{t \in T} ||\mathbb{E}S_{\text{real}}(t)S_{\text{real}}(t-\tau) - \mathbb{E}S_{\text{fake}}(t)S_{\text{fake}}(t-\tau)||dt$$

- **Квантили**. Нам важно следить за размером хвостов наших рядов, один из способов – сравнивать вектора квантилей.

$$Q_q = ||Q_q(S_{\text{real}}) - Q_q(S_{\text{fake}})||$$

- **Регуляризатор частоты колебаний**. Тут пришлось немного поломать голову, чтобы придумать, каким образом составить лосс, чтобы генератор штрафовался за генерацию слишком грубых, часто-колеблющихся траекторий.

Можно было добавить частоту колебаний как дополнительный признак, но не хотелось раздувать данные и препроцессить их, поэтому мы придумали еще один мета-лосс:

$$Osc(\delta) = \int\limits_{t \in T} \Big(S(t+\delta) - (S(t) + S'(t))\Big)^2 dt \approx \sum_{t \in discr(T)} \Big(S(t+1) - \big(S(t) + \delta(t)\big)\Big)^2$$

Идея проста, рассмотрим одномерный дискретный случай: пусть мы находимся в точке (t, S_t), рассмотрим, какой шаг предшествовал этой позиции: $S_t - S_{t-1}$. Добавим к нашей текущей точке такой же шаг: $S_t + (S_t - S_{t-1})$. Посмотрим, насколько это отличается от настоящего шага: $S_{t+1} - S_t + (S_t - S_{t-1})$. Если ряд менял направление, то полученная разница будет "большой так как будет приближаться суммой скорости прошлого шага и скорости следующего шага. Если же направления шагов совпадают, то метрика будет приближаться лишь разницей скоростей. Возьмём от полученных значений квадрат или модуль или что-либо еще, что убивает знак и просуммируем по всему временному ряду.

Как итог, получается метрика, которая заметно возрастает по мере роста частоты колебаний ряда. Теперь, чтобы сделать ряды похожими по гладкости на реальные достаточно потребовать похожие значения предложенной метрики.

3.d Данные

Нам было интересно обобщить модель авторов: взять больше временных рядов, генерировать больше точек. Для продвижения в этом направлении мы использовали данные по криптовалютным курсам: BTC-USD, ETH-USD, BNB-USD, USDT-USD. Частота данных – 1 день. Период – 5 лет.

4 Результаты

Для временных рядов нет каких-либо общепризнанных метрик качества, поэтому оценка результатов – отдельная задача. В этом разделе мы оценим, насколько получилось выполнить поставленную изначально задачу. Нам хочется, чтобы ряды сохраняли ковариационные зависимости между координатами, авторегрессионные зависимости и при этом адекватно выглядели.

4.a Борьба с mode collapse

Первая серьезная трудность – **mode collapse** сверточных сетей. К нашему удивлению, добиться того, чтобы сверточная архитектура авторов хотя бы на тех же данных и получить качество соразмерное обычному многослойнику – нетривиальная задача. Открытого репозитория у статьи не было, описание было достаточно обобщенным, поэтому все трюки по стабилизации приходилось изобретать самим и заново.

Мы пробовали варьировать количество шума, подаваемого модели на вход, мультистартовать с разных батчей, чтобы модель не скатывалась в коллапс на ранних стадиях, строить разные расписания обучения.

В итоге, оригинальные свертки не коллапсировали примерно 1 запуск из 7 и в дальнейшем, в качестве бейзлайна мы использовали в основном mlp.

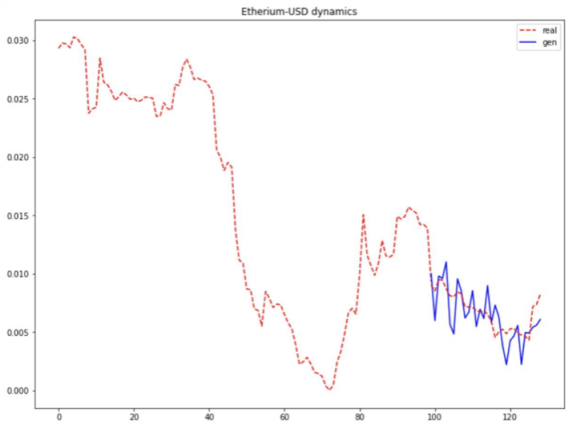
Многослойные сети и трансформеры уже не мучали нас коллапсами, поэтому переходим к следующей сложности.

4.b Решение проблемы с частотой и гладкостью

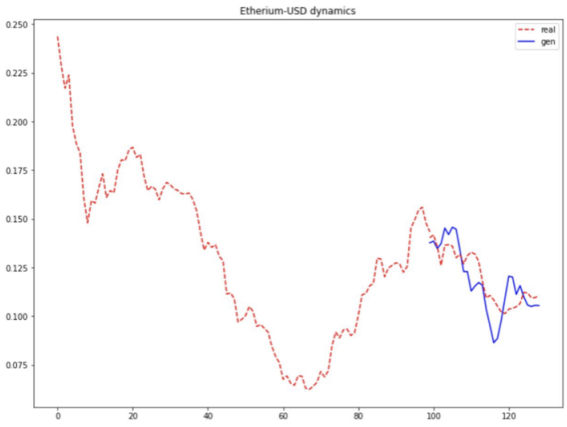
По ходу проекта мы заметили проблему, от которой регулярно страдают сети, генерирующие ряды: никак не учитывается частота колебаний ряда. В лоб это не решается, ибо посчитанная руками частота колебаний не продифференцируется торчем, поэтому пришлось изобретать.

Первой идеей было добавить количества колебаний в качестве признака, но очень не хотелось откланяться от фреймворка, в котором на вход подаются либо сами ряды, либо шум. Поэтому мы придумали $Osc(\delta)$, определенный и подробно описанный в разделе 3c.

Придуманный лосс хорошо себя проявил. Ряды стали выглядить похоже по гладкости на реальные данные, частота колебаний тоже выровнялась. При этом, косвенно регуляризовалась амплитуда колебаний и наказывались резкие скачки, если их не было в данных. С этим самостоятельно не справлялась ни одна архитектура дискриминатора и генератора из тех, что мы пробовали. Как это выглядит на удачно обученной сверточной сети:



Сверточная сеть без мета-признаков.



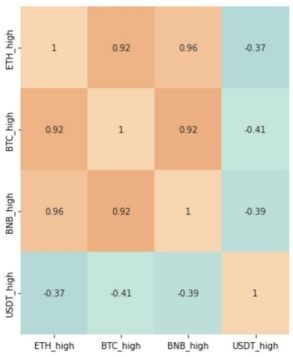
Та же архитектура, обученная с мета-признаками.

4.c Выявление созависимости между рядами

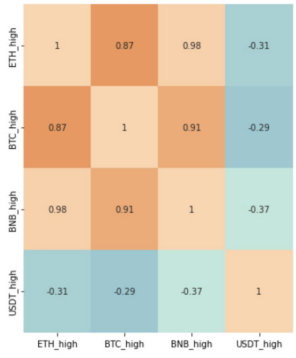
Мы добавляли в лосс близость ковариационной функции сгенерированной системы и реальных данных. Это, разумеется, обращает внимание модели на ковариации, но само по себе еще не значит, что модель будет робастна и все заработает на тестовых данных.

Сравним ковариационную матрицу сгенерированных рядов и реальных данных. Чтобы избежать не возникало вопросов по поводу cherry pick'a, будем использовать монте-карло с хотя бы 50 итерациями.

Эксперимент оказался удачным:



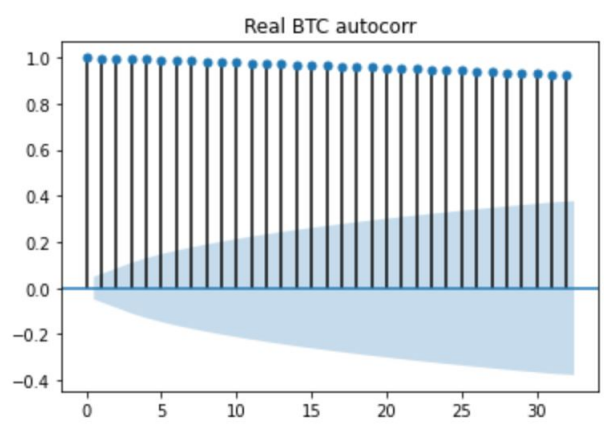
Корреляционная матрица исторических данных.

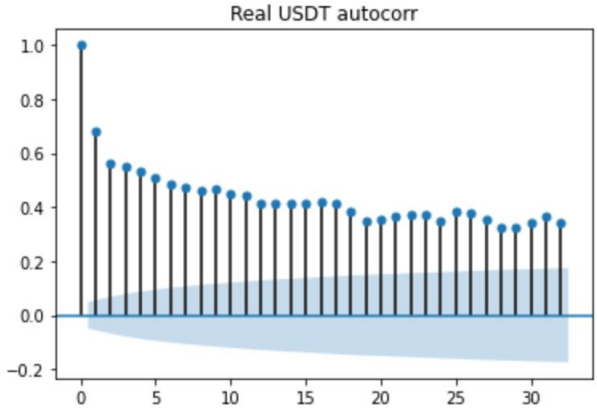


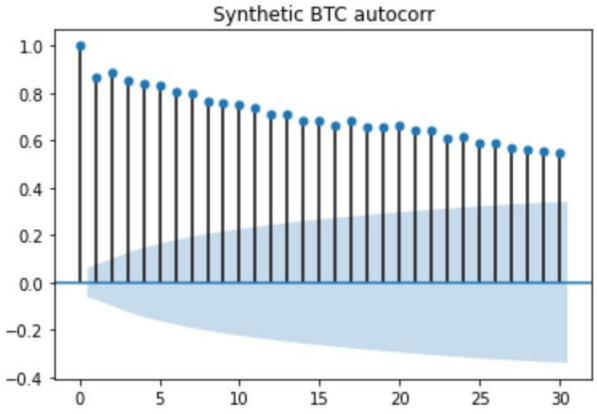
Корреляционная матрица сгенерированных данных.

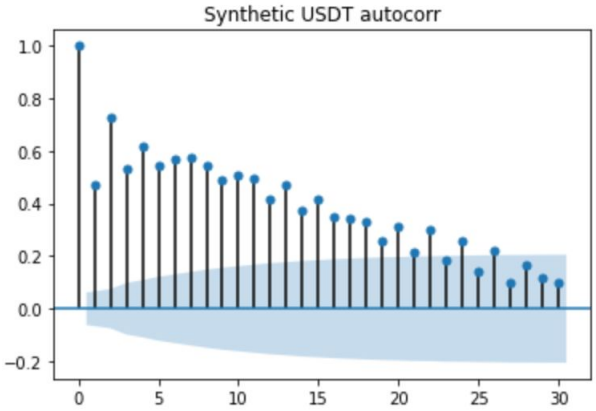
4.d Выявление автокорреляционных зависимостей

Еще одна субъективная мера качества. Мы бы хотели похожие автокорреляции, вроде как, мы их получили.



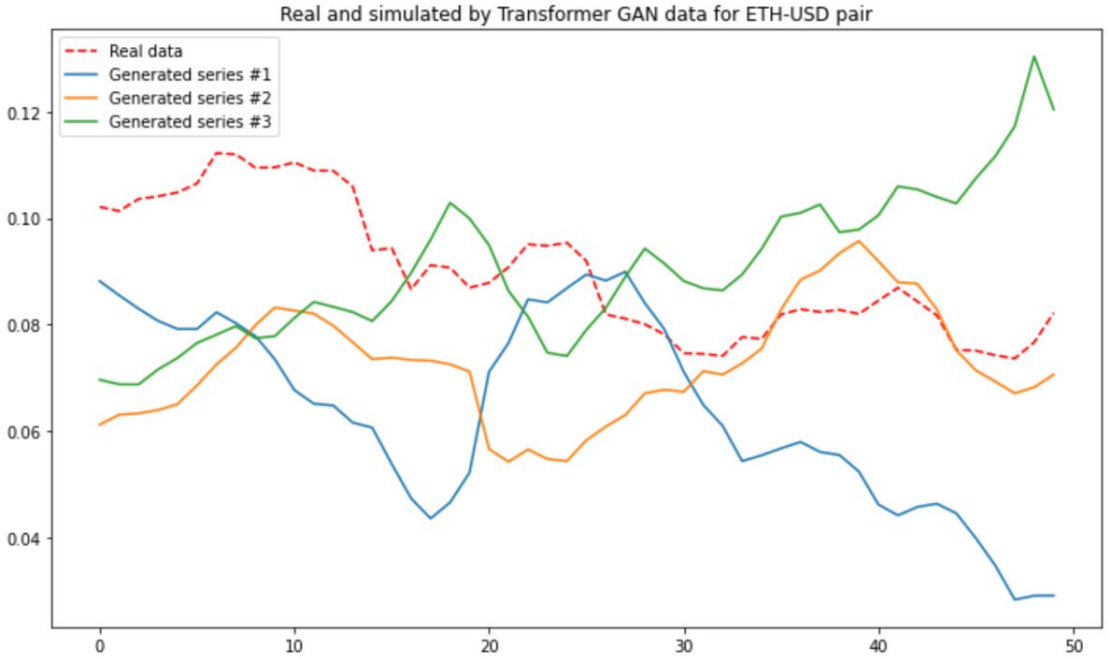


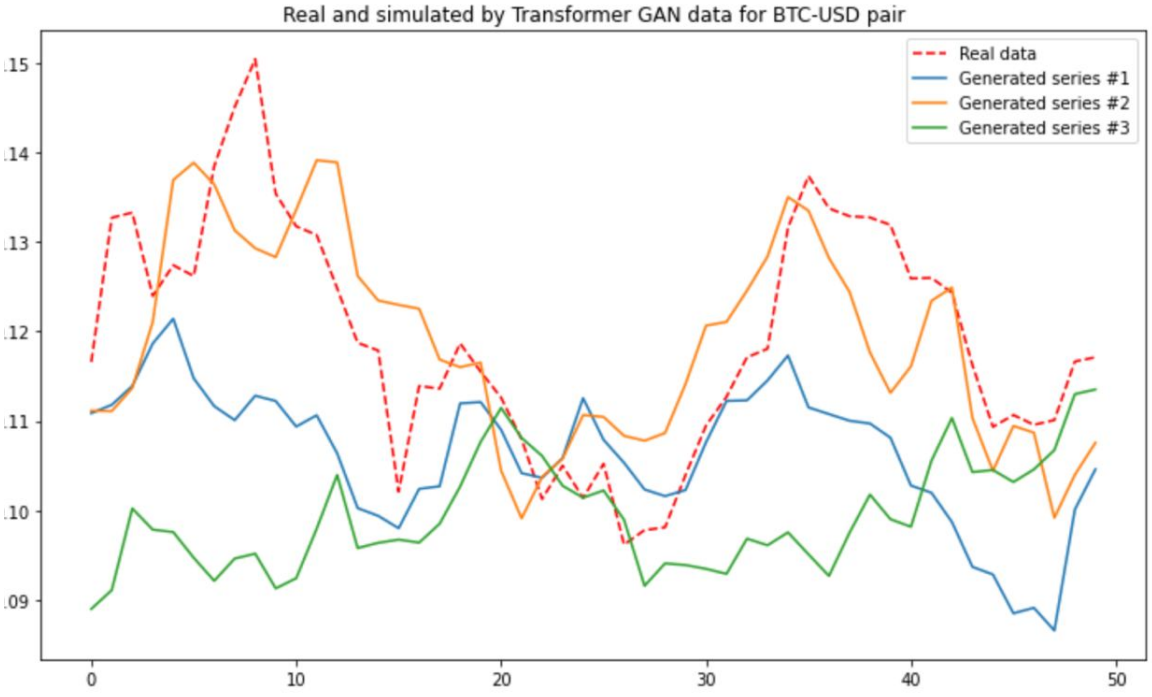


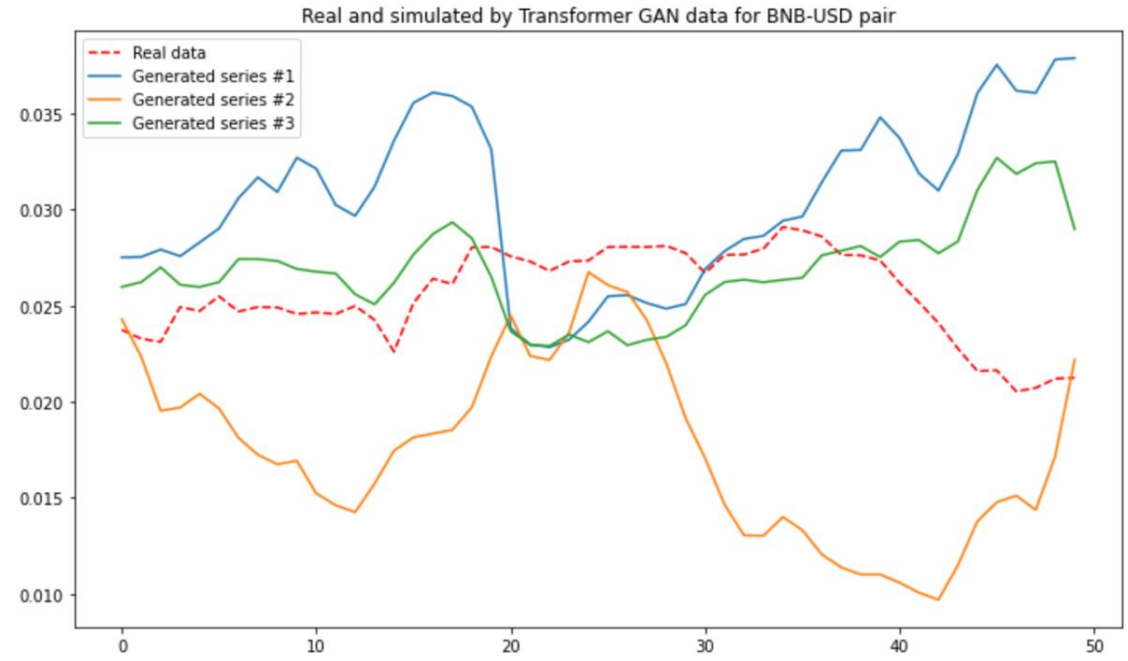


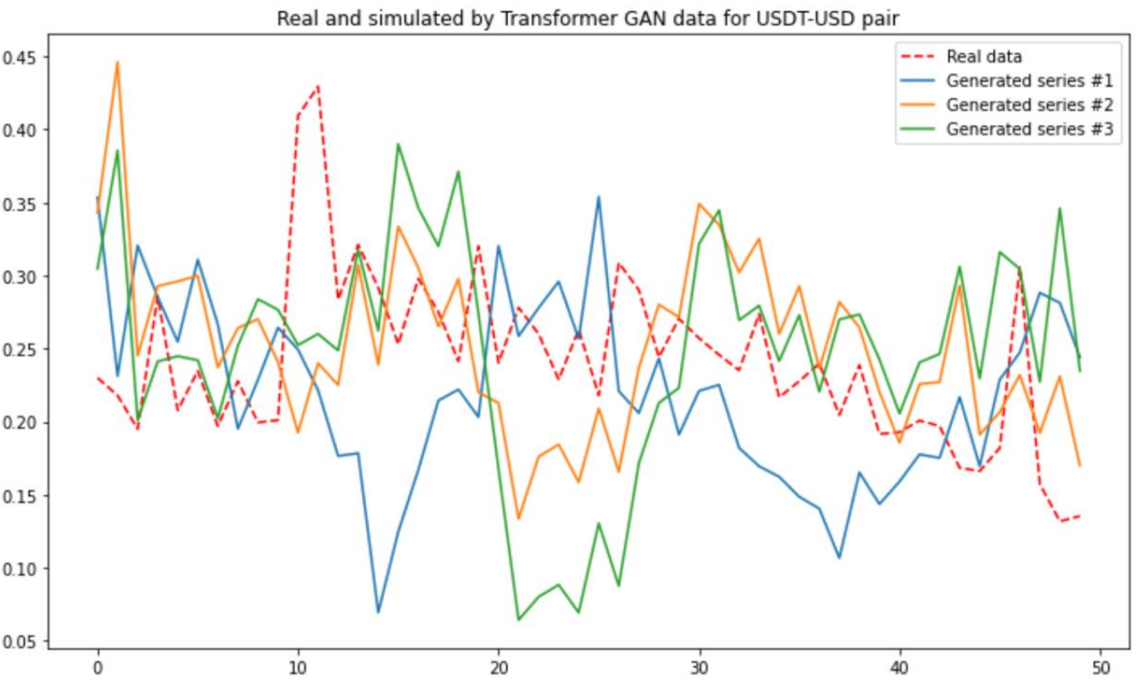
4.e Примеры сэмплов.

Мы долго держали интригу, наконец посмотрим на то, какие получаются ряды.






Real and simulated by Transformer GAN data for BNB-USD pair
Real data
Generated series #1
Generated series #2
Generated series #3
0.035
0.030
0.025
0.020
0.015
0.010
0
10
20
30
40
50


Real and simulated by Transformer GAN data for USDT-USD pair
Real data
Generated series #1
Generated series #2
Generated series #3
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
10
20
30
40
50