

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б08-мм

Генератор временных рядов с пространственными взаимосвязями

Кропачева Алена Михайловна

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
д.т.н., профессор кафедры системного программирования, Кознов Д. В.

Консультант:
старший инженер, ООО «Техкомпания Хуавей», Гирдюк Д. В.

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	6
2. Обзор	7
2.1. Основы временных рядов	7
2.2. Существующие решения	9
2.3. Используемые технологии	11
3. Модель генератора временных рядов	13
3.1. Стохастические процессы	13
3.2. Генератор расписания временных рядов	14
3.3. Способы сэмплирования исходных данных	16
3.4. Способы генерации параметров процессов	17
4. Реализация	20
4.1. Общее описание программной реализации	20
4.2. Сэмплирование исходных данных	20
4.3. Генерация параметров процессов	21
4.4. Архитектура	22
5. Эксперимент	25
5.1. Цель эксперимента	25
5.2. Метрики	25
5.3. Результаты	25
Заключение	28
Список литературы	30
Приложения	33

Введение

Временные ряды широко используются во многих областях — для предсказания финансовых показателей, анализа дорожного трафика, прогнозирования погоды, поиска аномалий в сетевом трафике [16] и др. За последние годы появилось большое количество открытых наборов данных, представляющих различные временные ряды, также проводятся масштабные соревнования по их предсказанию [11, 12]. Наблюдается увеличение интереса к сложным задачам по работе с временными рядами, например, иерархическое предсказание (*hierarchical forecasting*) [1] или моделирование пространственно-временных зависимостей (*spatiotemporal modeling*) [8].

Однако, в области анализа временных рядов имеется ряд проблем. Временные ряды, представленные во многих открытых наборах данных, достаточно однородны, что ограничивает их применение для оценки эффективности предсказательных моделей [10]. Кроме того, для определенных классов задач часто бывает сложно (или невозможно) найти открытые наборы данных [16]. В связи с этим методы генерации временных рядов могут помочь создать дополнительные данные, чтобы не только улучшить процесс обучения популярных на текущий момент глобальных нейросетевых моделей предсказания временных рядов, но и создать общие наборы данных (*benchmarks*), пригодные для их всестороннего тестирования. Примерами таких моделей являются как глобальные модели, например, *Informer*, *FEDformer*, *TFT*, так и базисные модели *TimeGPT*, *TimesNet*, *TimeLLM* (см. обзоры [17, 21]).

Методы генерации временных рядов могут быть разделены на две группы: статистические и нейросетевые [3]. В первой группе популярен авторегрессионный подход, методы второй группы чаще всего основаны на применении генеративных состязательных сетей (*GAN*) или вариационных автокодировщиков (*VAE*). Таким образом, в методах первой группы временные ряды получаются на основе параметризуемых авторегрессионных моделей, в методах второй группы нейросетевые модели обучаются генерировать временные ряды из случайного шума на основе

существующего набора данных. Первый подход реализован в таких инструментах, как GRATIS [10], timeseries-generator [22] и mockseries [19], а второй широко используется в областях, обладающих большим объемом реальных наблюдений, например, в сфере экономики и финансов [7].

Тем не менее, у подавляющего большинства существующих решений отсутствует поддержка генерации скоррелированных временных рядов, часто наблюдаемых в задачах моделирования пространственно-временных зависимостей между временными рядами и объектами, которые их порождают. Например, в задачах предсказания интенсивности дорожного трафика сенсоры, установленные на дорогах, замеряют количество проезжающих машин и/или среднюю скорость их потока. Соответствующие временные ряды между смежными перекрестками зачастую достаточно сильно коррелируют, что позволяет использовать эту информацию для улучшения качества предсказаний, например, используя глобальные графовые нейронные модели (spatiotemporal graph neural networks, STGNN [6]). Появление открытых наборов данных дорожного трафика, таких как PEMS-BAY и METR-LA [8], позволило исследователям унифицировать способ сравнения своих моделей, что, в свою очередь, значительно ускорило прогресс в развитии моделирования пространственно-временных взаимосвязей. Тем не менее, подобный тип взаимосвязей встречается во множестве других областей, таких как телекоммуникации, нейробиология, эпидемиология, метеорология и др. Но на текущий момент во многих областях либо вовсе отсутствуют открытые наборы данных, либо они незначительны по объему, что еще раз подтверждает необходимость разработки методов генерации временных рядов.

Кроме того, анализ временных рядов не ограничивается лишь их предсказанием. Поиск аномалий как в отдельных наблюдениях, так и аномальных временных рядов, задачи поиска причинно-следственных связей, адаптивное предсказание временных рядов, допускающих смену распределения значений представляют в настоящий момент значительный интерес.

В данной работе представлен метод генерации временных рядов с пространственными взаимосвязями, порождаемых некоторым множеством объектов (например, сенсорами на дорогах или базовыми станциями сотовой связи). Временные ряды генерируются с помощью набора классических параметризуемых авторегрессионных моделей, например, ETS [2]. Пространственные взаимосвязи моделируются путем сэмплирования параметров процессов: близость объектов, определяемая, например, по евклидовым или геодезическим расстояниям на некотором многообразии в пространстве параметров, соответствует близости параметров авторегрессионных моделей. Чем ближе параметры процессов, тем более похожи временные ряды друг на друга.

Идея данного проекта и соответствующее задание для учебной практики были предоставлены сотрудниками команды математического моделирования и оптимизации техкомпании «Хуавэй».

1 Постановка задачи

Целью работы является разработка приложения для генерации наборов данных временных рядов с пространственными взаимосвязями. Для её выполнения были поставлены следующие задачи:

1. Выполнить обзор существующих решений для генерации временных рядов;
2. Реализовать набор процессов для генерации комплексных моделей временных рядов;
3. Реализовать различные способы сэмплирования исходных данных;
4. Реализовать различные способы генерации параметров процессов на основе исходных данных;
5. Провести тестирование генератора временных рядов;

2 Обзор

2.1 Основы временных рядов

Временной ряд — это набор наблюдений $y_1, y_2, \dots, y_t$, каждое из которых было получено в определенный момент времени t . Модель временного ряда:

$$Y_{t+d}(w) = f_{t,d}(y_1, \dots, y_t; w; \xi)$$

где d — горизонт прогнозирования, w — вектор параметров модели, ξ — компонента случайной ошибки [23].

Модели, реализуемые в генераторе временных рядов:

1. Белый шум (white noise):

$\{X_t\} \sim \text{IID } N(0, \sigma^2)$, где σ^2 - дисперсия, N - нормальное распределение.

Параметры модели: σ^2

Наиболее простая модель временного ряда, в которой все наблюдения независимы и нормально распределены со средним значением 0. Белый шум редко используется в качестве самостоятельной и полноценной модели для прогнозирования временного ряда, но является аддитивной или мультипликативной компонентой во многих комплексных моделях [2].

2. Случайное блуждание:

(a) Простое симметричное случайное блуждание (simple symmetric random walk):

$$Y_t = X_1 + X_2 + \dots + X_t, \text{ for } t = 1, 2, \dots,$$

$$P[X_i = 1] = p, P[X_i = -1] = 1 - p$$

P - вероятность, $p \in [0, 1]$

Параметры модели: p, X_0

В модели простого симметричного случайного блуждания текущее значение Y_t вычисляется как $Y_t = Y_{t-1} + 1$ с вероятностью p или $Y_t = Y_{t-1} - 1$ с вероятностью $1 - p$ [2].

(b) Случайное блуждание (random walk):

$$Y_t = X_1 + X_2 + \dots + X_t, \text{ for } t = 1, 2, \dots,$$

$$\{X_t\} \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: σ^2, Y_0

В модели простого симметричного случайного блуждания текущее значение Y_t вычисляется как $Y_t = Y_{t-1} + X_t$, где $\{X_t\}$ - независимы и одинаково распределены со средним значением 0 [2].

3. Простое экспоненциальное сглаживание

(simple exponential smoothing)

$$Y_t = L_{t-1} + U_t$$

$$L_t = L_{t-1} + \alpha \cdot U_t, \alpha \in [0, 1]$$

$$U_t \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: α, σ^2, L_0

В модели простого экспоненциального сглаживания предсказываемое наблюдение Y_t зависит от долгосрочного равновесного значения L_t и независимо и нормально распределенной ошибки U_t , взятой с коэффициентом α [9].

4. Метод Хольта (double exponential smoothing)

$$Y_t = L_{t-1} + B_{t-1} + U_t$$

$$L_t = L_{t-1} + B_{t-1} + \alpha \cdot U_t, \alpha \in [0, 1]$$

$$B_t = B_{t-1} + \beta \cdot U_t, \beta \in [0, 1]$$

$$U_t \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: $\alpha, \beta, \sigma^2, L_0, B_0$

Метод Хольта расширяет модель простого экспоненциального сглаживания, добавляя компоненту тренда, которая отражает тенденцию к росту или падению значений временного ряда. За выраженность тренда в модели отвечает коэффициент β [9].

5. Метод Хольта-Уинтерса (triple exponential smoothing)

$$Y_t = L_{t-1} + B_{t-1} + S_{t-l} + U_t$$

$$L_t = L_{t-1} + B_{t-1} + \alpha \cdot U_t, \alpha \in [0, 1]$$

$$B_t = B_{t-1} + \beta \cdot U_t, \beta \in [0, 1]$$

$$S_t = S_{t-l} + \gamma \cdot U_t, \gamma \in [0, 1], l \in N, l \leq t$$

$$U_t \sim \text{iID } N(0, \sigma^2)$$

Параметры модели: $\alpha, \beta, \gamma, \sigma^2, L_0, B_0, S_0, S_1, \dots, S_{l-2}$

Метод Хольта-Уинтерса расширяет метод Хольта путем добавления компоненты сезонности, которая отображает появление во временном ряде некоторого паттерна поведения значений с известной фиксированной частотой. За выраженность сезонности в модели отвечает коэффициент θ [9].

2.2 Существующие решения

Методы генерации временных рядов можно разбить на две группы: статистические (авторегрессионные) и нейросетевые. Обзор современных нейросетевых подходов к генерации временных рядов представлен в работе [3]. Описываемый в данной статье метод относится к классу авторегрессионных подходов генерации временных рядов. Рассмотрим ряд существующих методов и их реализаций из этой группы.

Метод генерации временных рядов с разнообразными контролируемые характеристиками GRATIS[10] использует Gaussian mixture autoregressive (MAR) models. Настройка параметров этих моделей позволяет создавать ряды с желаемыми особенностями. GRATIS генерирует реалистичные наборы данные, которые необходимы для таких исследовательских задач, как сравнение предсказаний, обучение модели на сгенерированных данных и т.д. Таким образом, метод предоставляет эффективный инструмент сравнительного анализа, но не рассматривает генерацию набора пространственно взаимосвязанных временных рядов.

В [4] представлен метод, который генерирует синтетические временные ряды, применимые в задачах симуляции, контроля и оптимизации гибридных энергетических систем (hybrid energy systems, HES). Для генерации временных рядов используются ряды Фурье и ARMA-модели, обученные на реальных измерениях. Помимо генерации независимых

рядов, метод предлагает способ получения синтетических коррелирующих рядов на основе коррелирующих входных данных. Однако, этот метод лишь частично удовлетворяет целям работы, так как в нем корреляция временных рядов полностью определяется данными для обучения и применяется для всей результирующей системы, в то время как концепция пространственной взаимосвязи подразумевает корреляцию только тех временных рядов, которые порождены близкими объектами.

Помимо описанных выше методов имеются программные инструменты для генерации временных рядов. Многие из них являются библиотеками (в частности, на Python), которые порождают временные ряды с помощью различных стохастических процессов. Процесс в данном случае — это некоторая функция, описывающая поведение значений ряда в зависимости от времени. Однако существующие решения не поддерживают создание набора временных рядов с учетом пространственных взаимосвязей, представляя интерес лишь с точки зрения реализованных процессов, методов генерации и программной архитектуры. В данной статье рассмотрены два таких пакета: *timeseries-generator* и *mockseries*.

Широко известен Python-пакет *timeseries-generator*, предоставляющий возможность генерировать синтетические временные ряды. Временной ряд в данной реализации представляется в следующем виде: $ts = v_0 \cdot f_1 \cdot f_2 \cdot \dots \cdot f_N + u$, где v_0 обозначает начальное значение, $f_1, \dots, f_N$ — целочисленные факторы, отражающие характеристики временного ряда (тренд, сезонность и т.д.), а u есть случайная ошибка (шум) [22]. Этот инструмент поддерживает возможность задания различных особенностей временного ряда, таких как тренд и сезонность, а также позволяет симулировать изменение наблюдений, связанное с реальными событиями (выходные, праздники, “чёрная пятница” и т.п.). Тем не менее, данный пакет не поддерживает возможности параметризации стохастических процессов набором данных извне, и для генерации временных рядов используется только одна модель, заданная пользователем. Это означает, что исследование полученных синтетических временных

рядов будет достаточно тривиальной задачей, так как отсутствуют точки изменения характеристик. Таким образом, пакет `timeseries-generator` содержит эффективные методы генерации временных рядов, но не удовлетворяет всем поставленным задачам.

Гибкие возможности по созданию временных рядов предоставляет другой Python-пакет — `mockseries` [19]. Он имеет больше компонент, чем `timeseries-generator`, и представляет временные ряды как аддитивные и мультипликативные комбинации различных сигналов (тренд, сезонность, шум и т.п.), например: $y_t = (S_1(t) \cdot T_l(t) + S_2(t)) \cdot u$, где через $S_1(t)$, $S_2(t)$ обозначены различные сезонные компоненты, $T_l(t)$ есть линейный тренд, а u — случайная ошибка (шум) [24]. Кроме того, пакет имеет методы для изменения значений временного ряда в указанный момент времени либо на заданном промежутке времени для имитации аномалий, а также может изменять значения временного ряда на заданном промежутке в соответствии с определенной функцией. Таким образом, с помощью инструмента `mockseries` можно сгенерировать набор временных рядов, характеристики которых изменяются с течением времени. Однако пакет не решает проблему параметризации процессов некоторым исходным набором данных.

Подводя итог, можно сказать, что в литературе отсутствуют методы генерации временных рядов, порожденных объектами с пространственными взаимосвязями, также нет соответствующих программных инструментов.

2.3 Используемые технологии

Во время работы над генератором временных рядов использовались:

- Python — высокоуровневый язык программирования, обладающий эффективным подходом к объектно-ориентированному программированию [15].
- NumPy — Python-библиотека для научных вычислений [14].

- `matplotlib` — Python-библиотека для создания статичных, анимированных и интерактивных визуализаций [18].
- `scikit-learn` — Python-библиотека, которая предоставляет различные методы машинного обучения, анализа данных и др [20].
- `NetworkX` — Python-библиотека для создания, управления и изучения структуры графовых сетей [13].

3 Модель генератора временных рядов

Модель генератора временных рядов включает в себя следующие компоненты:

- Набор стохастических процессов, поддерживающих параметризацию.
- Генератор “расписания” — механизм создания комплексных моделей временных рядов.
- Генератор исходных данных.
- Генератор параметров процессов.

Для вычисления различных параметров и начальных значений модель должна быть определена в пространстве, которое задается пользователем или вычисляется в зависимости от исходных данных. Обозначим текущее пространство модели как пару $([a, b], m)$, $[a, b] \in R$, $m \in N$, где $[a, b]$ — промежуток значений, а m — количество разбиений промежутка. Тогда все значения X_t , необходимые для инициализации процессов, будут некоторым образом выбраны из $[a, b]$: $X_T \in [a, b]$, а параметр стандартного отклонения будет зависеть от a, b, m : $\sigma = f(\frac{b-a}{m})$.

3.1 Стохастические процессы

Функции, которые определяют поведение временного ряда и содержат компоненту случайной ошибки, называются стохастическими процессами. Они описывают рост, убывание, стационарность, сезонность, дисперсию и другие характеристики ряда. Процессы позволяют вычислить новое значение ряда на основе нескольких предыдущих. Временные ряды, созданные одинаковой последовательностью процессов, схожи между собой.

В генераторе временных рядов процесс представлен следующим образом: $p = f(X_t, X_{t-1}, \dots, X_{t-l}, \xi; \theta)$, где $X_t, X_{t-1}, \dots, X_{t-l}$ — набор предыдущих значений ряда, θ — вектор параметров, ξ — компонента случайной

ошибки [23]. Таким образом, временной ряд оказывается последовательностью значений некоторого набора процессов. Если процесс запускается первым, то он получает сгенерированный набор данных, иначе он использует предыдущие значения временного ряда.

Для того, чтобы сгенерированные временные ряды были похожи на реальные данные, в методе используются ETS-модели (Error, Trend and Seasonality). В реализации генератора временных рядов все ETS-модели — аддитивные, они представлены в виде суммы нескольких компонент:

$$ts = a \cdot L(t) + b \cdot T(t) + c \cdot S_l(t) + d \cdot U(t),$$

где через $L(t)$ обозначена долгосрочная компонента, $T(t)$ — тренд, $S_l(t)$ — сезонная компонента с частотой l , а $U(t)$ есть случайная ошибка [2]. Параметры $a, b, c, d \in N$ обозначают количество компонент соответствующего вида, которые задаются пользователем.

3.2 Генератор расписания временных рядов

Расписание — это порядок процессов, каждому из которых поставлена в соответствие последовательность параметров. При этом порядок процессов оказывается последовательностью пар $(process_i, steps_i)$, где $process_i$ — случайно выбранный процесс из списка доступных процессов, а $steps_i \in N$ — количество шагов, отведенное процессу. Пусть k задает количество пар в последовательности, а n — общее число наблюдений временного ряда. Тогда имеем $\sum_{i=1}^k steps_i = n$.

Для каждой пары $(process_i, steps_i)$ сгенерируем последовательность, состоящую из пар вида $(steps_{ij}, parameters_{ij})$, где $parameters_{ij}$ — набор параметров процесса, а $steps_{ij}$ — это количество шагов, отведенное процессу для работы с этим набором параметров. Если q определяет количество пар в последовательности параметров, то имеем $\sum_{j=1}^q steps_{ij} = steps_i$. На Рис. 1 представлена визуализация расписания временных рядов.

Example of time series generated by schedule

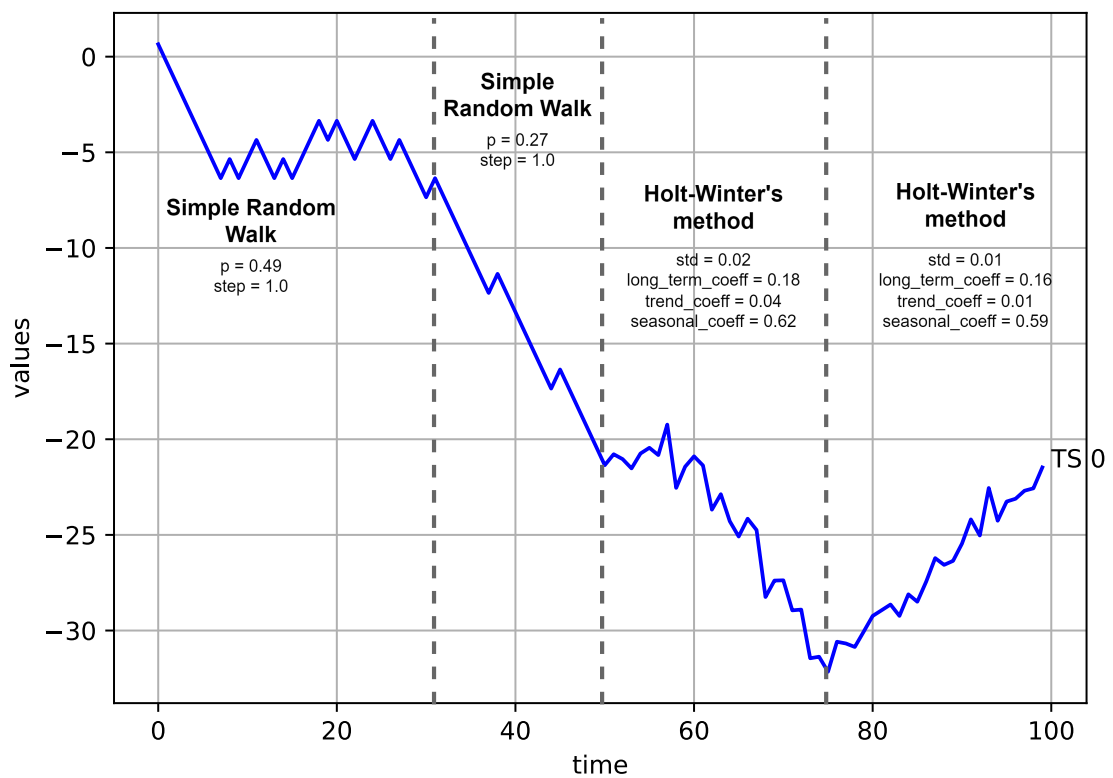


Рис. 1: Визуализация расписания

3.3 Способы сэмплирования исходных данных

Первый способ генерации исходных данных — сэмплирование точек на произвольной поверхности второго порядка. Точки являются рассматриваемыми объектами, а их координаты — характеристиками. Для оценки близости точек в пространстве можно использовать евклидово или геодезическое расстояние. Евклидово расстояние отражает территориальную близость, а геодезическое позволяет моделировать более сложные отношения.

Рассмотрим тривиальный способ сэмплирования точек на поверхности второго порядка. Пусть $B \in R^3$ — сфера с центром в начале координат и радиусом r . Назовем точку $(x, y, z) \in R^3$ объектом, если $(x, y, z) \in B$. Тогда множество объектов O можно задать следующим образом: $O = \{(x, y, z) : (x, y, z) \in B\}$. В качестве отношения на множестве O возьмем евклидово расстояние R : $R((x_1, y_1, z_1), (x_2, y_2, z_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. Таким образом, чем меньше R , тем ближе объекты и порождаемые ими временные ряды.

После сэмплирования точек выполняется кластеризация объектов. Она позволяет реализовать следующую идею параметризации комплексной модели временного ряда: для каждого кластера генерируется собственное случайное расписание, и все объекты порождают временные ряды, используя расписание своего кластера. В данной реализации номер кластера — это единственный параметр, от которого зависит генерация расписания.

Второй способ создания исходных данных — генерация графа. В графе объектами являются вершины, а отношения между объектами задают ребра: вершины, соединенные ребром, являются близкими между собой. Если визуализировать граф на плоскости или поверхности, то в качестве характеристик объектов можно взять координаты вершин — в таком случае с графом можно работать как с набором обычных точек.

На Рис. 2 показана схема сэмплирования исходных данных на поверхности и генерация графа.



Генерация исходных данных в виде графа



Рис. 2: Схема сэмплирования исходных данных

3.4 Способы генерации параметров процессов

Генерация параметров позволяет временному ряду зависеть от характеристик родительского объекта. Таким образом, если характеристики объектов схожи, то их временные ряды будут порождаться одинаковыми наборами процессов с близкими параметрами. Основной проблемой параметризации является преобразование характеристик в параметры процессов: число характеристик и параметров может не совпадать, параметры процессов почти всегда должны находиться в заданном интервале и т.д. Для решения этой проблемы в методе рассмотрены следующие подходы:

- агрегация характеристик,
- сопоставление пространства характеристик с пространством параметров.

Первый подход предполагает агрегацию характеристик исходных данных и использование полученного значения для генерации всех параметров процессов. Визуализация метода представлена на Рис. 3.

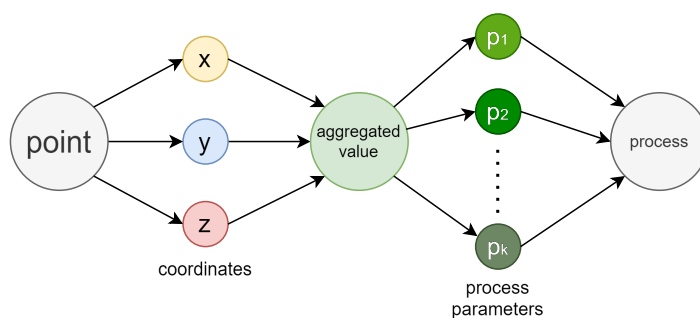


Рис. 3: Схема агрегации

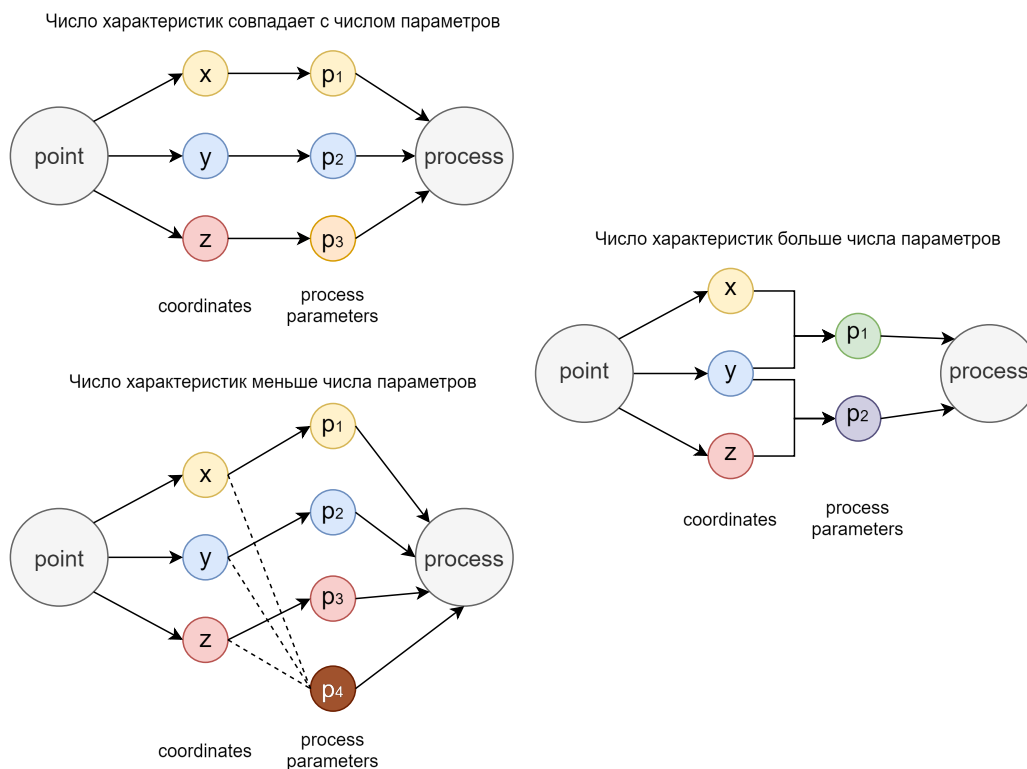


Рис. 4: Схема сопоставления пространств характеристик и параметров

Второй подход сопоставляет пространство характеристик объекта пространству параметров процесса. Если число характеристик и параметров совпадает, то все характеристики по порядку преобразуются к диапазону соответствующего параметра. Если число параметров превышает число характеристик, то пространство характеристик пополняется агрегированными значениями. Если параметров меньше, чем характеристик, то характеристики заменяются агрегированными значениями. Визуализация метода представлена на Рис. 4.

Подобранные параметры могут привести к неустойчивости времен-

ных рядов, поэтому для решения этой проблемы пользователю необходимо задать промежуток $[a, b]$ и число разбиений промежутка m , либо подать на вход набор данных, по которым можно рассчитать данные параметры. Например, если программа принимает набор координат точек в пространстве $\mathbb{R}^3$, то в качестве параметра a берется минимальное значение по всем координатам, а в качестве параметра b — максимальное. Эти ограничения используются в методе для генерации параметров и начальных значений временного ряда.

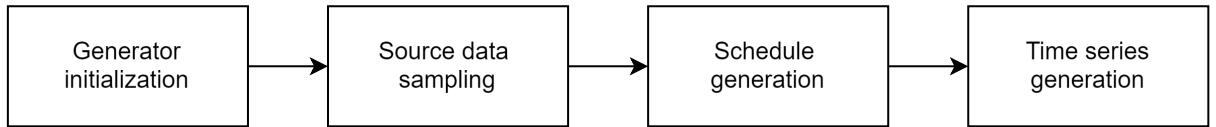


Рис. 5: Схема генерации временных рядов с сэмплированием исходных данных на сфере

4 Реализация

4.1 Общее описание программной реализации

Реализация приложения была разбита на две части: создание генератора временных рядов, поддерживающего комплексные пользовательские или случайные модели, и внедрение параметризации стохастических процессов, порождающих временные ряды, на основе исходных данных. На Рис. 5 представлена схема генерации временных рядов.

Для того, чтобы избежать неустойчивости временных рядов, пользователю необходимо задать промежуток $[a, b]$ и число разбиений промежутка m , либо подать на вход набор данных, по которым можно рассчитать данные параметры. Например, если программа принимает набор координат точек в пространстве R^3 , то в качестве параметра a берется минимальное значение по всем координатам, а в качестве параметра b — максимальное. Программа использует эти ограничения для генерации произвольных параметров процессов и начальных значений временного ряда.

4.2 Сэмплирование исходных данных

Все способы сэмплирования исходных данных наследуются от абстрактного класса *SourceDataSamplingMethod*. Этот класс имеет методы сэмплирования и визуализации данных, которые переопределяют конкретные реализации (*SurfaceSamplingMethod* и *GraphSamplingMethod*). *SourceDataSamplingMethod* возвращает набор объектов класса *SourceData*, который хранит характеристики и граф отношений данных. В случае

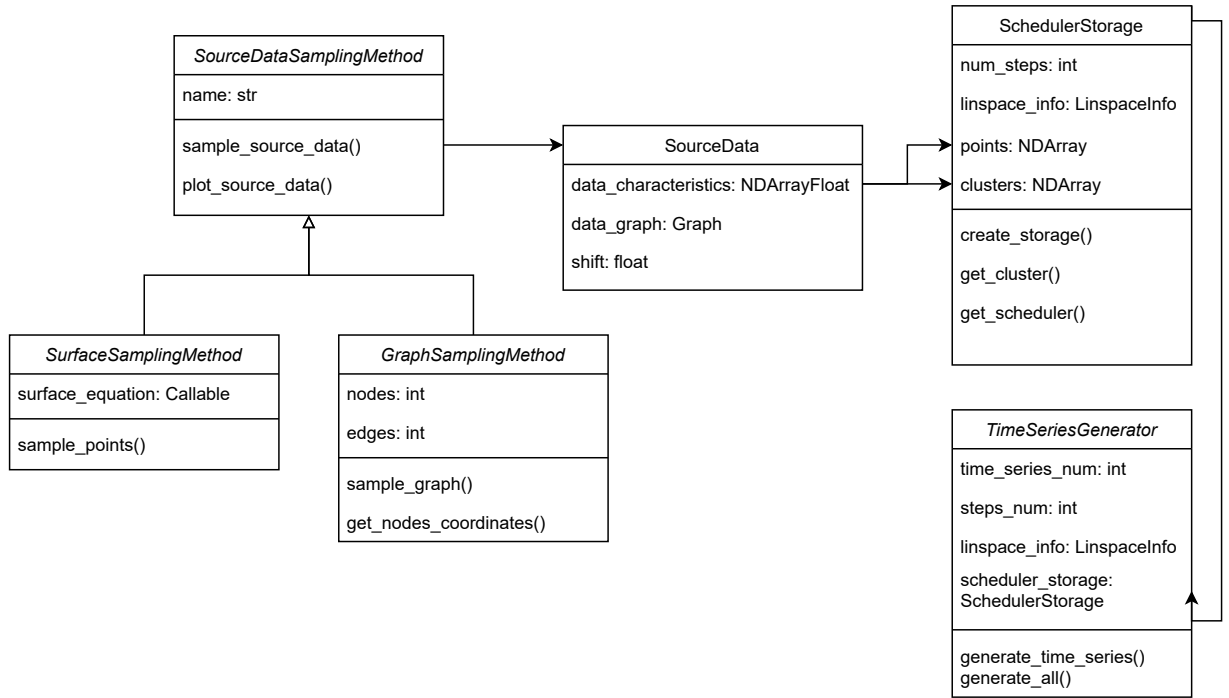


Рис. 6: Архитектура методов сэмплирования исходных данных

сэмплирования точек на произвольной поверхности граф остается пустым. После выполнения кластеризации объекты, соотнесенные с кластерами, попадают в *ScheduleStorage*, где для каждой области близких объектов генерируется уникальное расписание, затем по расписаниям создаются временные ряды. Архитектура методов сэмплирования исходных данных представлена на Рис. 6.

4.3 Генерация параметров процессов

Все способы параметризации процессов являются наследниками абстрактного класса *ParametersGenerationMethod*. Реализация каждого способа имеет методы генерации различных видов параметров: параметров, похожих на значения временного ряда (*MeanType*), параметров с нормальным распределением (*StdType*) и параметров с равномерным распределением, принадлежащих некоторому интервалу (*CoefficientType*). Метод генерации передается в *ProcessStorage* для инициализации процессов, а затем используется при генерации расписания в методах класса *TimeSeriesGenerator*. Архитектура методов параметризации процес-

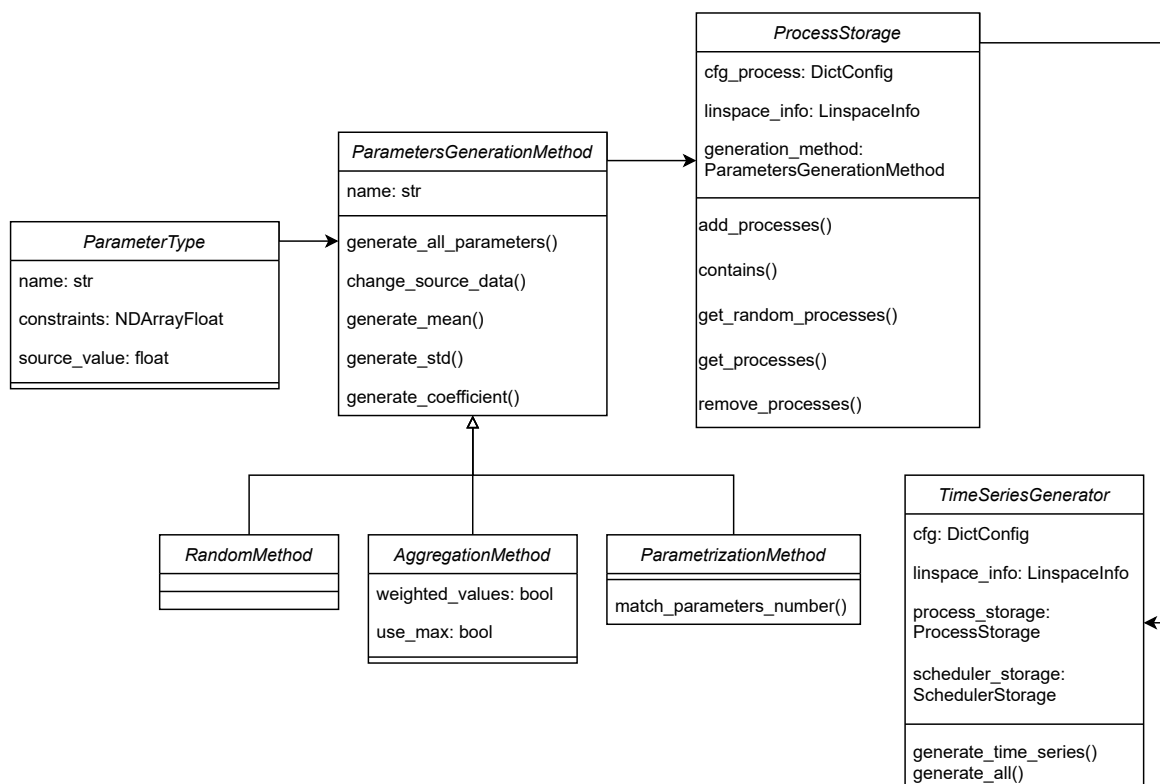


Рис. 7: Архитектура методов параметризации процессов

сов представлена на Рис. 7.

4.4 Архитектура

Приложение реализовано на языке Python [15] с использованием библиотеки NumPy [14], которая предоставляет эффективные методы работы с массивами и матрицами данных. Генерация и кластеризация точек на сфере выполнена с помощью пакета scikit-learn [20], для визуализации точек и временных рядов была использована библиотека matplotlib [18].

UML-диаграмма архитектуры генератора временных рядов представлена в приложении 1 (рис. 10). Описание архитектуры приложения:

1. TimeSeriesGenerator — интерфейс самого высокого уровня, имеет метод `generate_time_series()`, который генерирует один временной ряд в соответствии с расписанием, и `generate_all()`, который задает расписания и возвращает набор временных рядов.

2. `LinspaceInfo` — класс, содержащий гиперпараметры пользователя a, b, m , определенные в секции 4.1. Содержит метод `generate_values()`, который выдает набор случайных значений из промежутка $[a, b]$, и метод `generate_std()`, который генерирует случайное стандартное отклонение.
3. `SchedulerStorage` — класс, предназначенный для создания и хранения расписаний, соотнесенных с кластерами исходных данных. Метод `create_storage()` создает словарь вида $\{cluster : scheduler\}$, а `get_scheduler()` позволяет получить `Scheduler` по номеру кластера.
4. `TimeSeries` — класс, содержащий значения временного ряда и метаданные о процессах, использованных для генерации этого временного ряда.
5. `Scheduler` — класс, реализующий расписание (см. пункт 4.1.3) временного ряда. Имеет методы `generate_process_order()`, возвращающий объект `ProcessOrder`, и `generate_schedule()`, возвращающий расписание.
6. `Process` — класс, реализующий общий интерфейс стохастических процессов. Для инициализации требует `BaseParametersGenerator` и `ParametersGenerationMethod`. Имеет метод `generate_time_series()`, который возвращает временной ряд, и метод `get_info()`, возвращающий информацию о процессе. Все реализации конкретных процессов наследуются от `Process`.
7. `BaseParametersGenerator` — класс, реализующий общий интерфейс генератора параметров и начальных значений процессов. Необходим для инициализации `Process`. Имеет методы `generate_parameters()` и `generate_init_values()`. Все реализации `BaseParametersGenerator` для конкретных процессов значений процессов наследуются от `BaseParametersGenerator`.

8. `ProcessList` — класс, предназначенный для хранения списка доступных процессов. Имеет операции взаимодействия со списком: добавление процессов - `add_processes()`, возвращение процессов по названию - `get_processes()`, удаление процессов - `remove_processes()`, возвращение набора случайных процессов - `get_random_processes()`, проверка на наличие процесса в списке - `contains()`.

5 Эксперимент

5.1 Цель эксперимента

Целью эксперимента является проверка результатов работы генератора и оценка степени соответствия между временными рядами и породившими их точками на сфере.

5.2 Метрики

Критерием качества генератора временных рядов является схожесть временных рядов, порожденных близкими объектами. Для поиска оптимальных соответствий между временными последовательностями используют алгоритм динамической трансформации временной шкалы (DTW-алгоритм), он эффективно работает при сравнении временных рядов, один из которых смещен, сжат или растянут по временной шкале относительно другого. Однако, такой алгоритм не подходит для данной реализации генератора, так как полученные временные ряды могут отличаться динамикой и направлением тренда, даже если в их основе лежит одна и та же модель.

Альтернативный подход оценки качества — имитация реальных данных и использование искусственных данных для предварительного обучения моделей прогнозирования временных рядов. В настоящее время он используется для экспериментов с существующими наборами данных, обладающими пространственно-временными отношениями, такими как PEMS-BAY [5]. Идея подхода заключается в том, чтобы сравнить метрики моделей, обученных на реальных данных с пространственно-временными отношениями и синтетических данных, моделирующих реальные.

5.3 Результаты

На Рис. 8 слева находится график временных рядов, возвращенных генератором, а справа расположение на сфере точек, которые породили

временные ряды. Точки, принадлежащие одному кластеру, и соответствующие им временные ряды отмечены одинаковым цветом.

Временные ряды, порождаемые одним расписанием (кластером) явно различимы, они имеют схожую дисперсию и общие участки, где изменяются характеристики процессов и сами стохастические процессы. Чем ближе точки внутри кластера, тем ближе их начальные значения и порождаемые временные ряды: это отображено на тех графиках, где есть три временных ряда из одного кластера, — два ряда близки, а третий отличается от пары, но все еще сохраняет общую модель поведения.

Временные ряды из разных кластеров порождаются разными наборами процессов, поэтому их поведение слабо коррелирует, участки изменения не совпадают.

Тем не менее, схожесть временных рядов с учетом координат точек варьируется при каждой генерации. В некоторых наборах появляются схожие временные ряды независимо от близости их родительских объектов. Таким образом, метод генерации показал себя перспективным, но для его использования требуется провести дополнительное тестирование.

Помимо этого были проведены эксперименты по генерации синтетических данных, похожих на реальные. В них был использован датасет REMS-BAY [5], содержащий данные с сенсоров дорожного движения. На Рис. 9 зеленым показаны коррелирующие данные с разных сенсоров, а синим и оранжевым отмечены синтетические временные ряды.

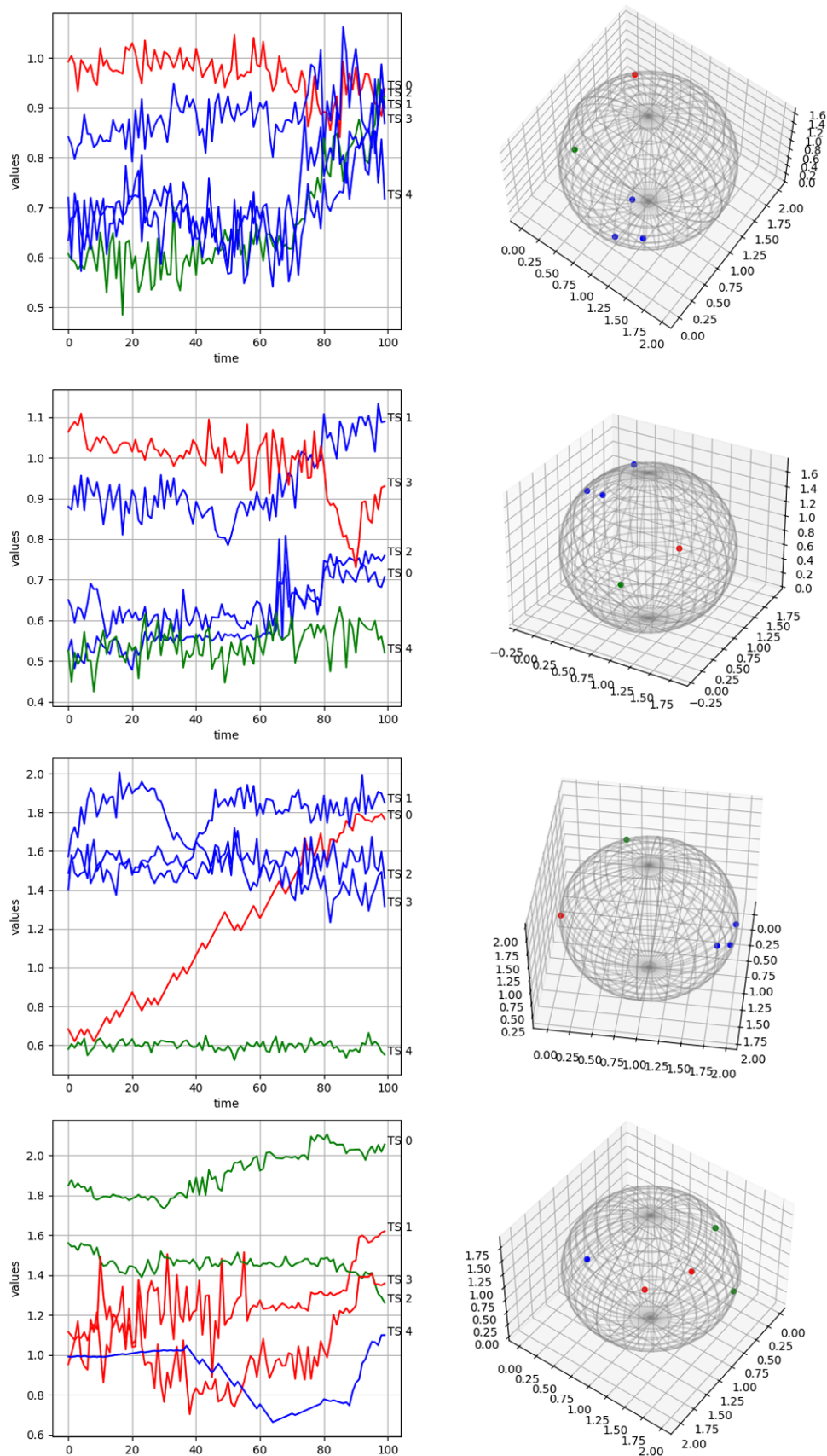


Рис. 8: Результат работы генератора временных рядов

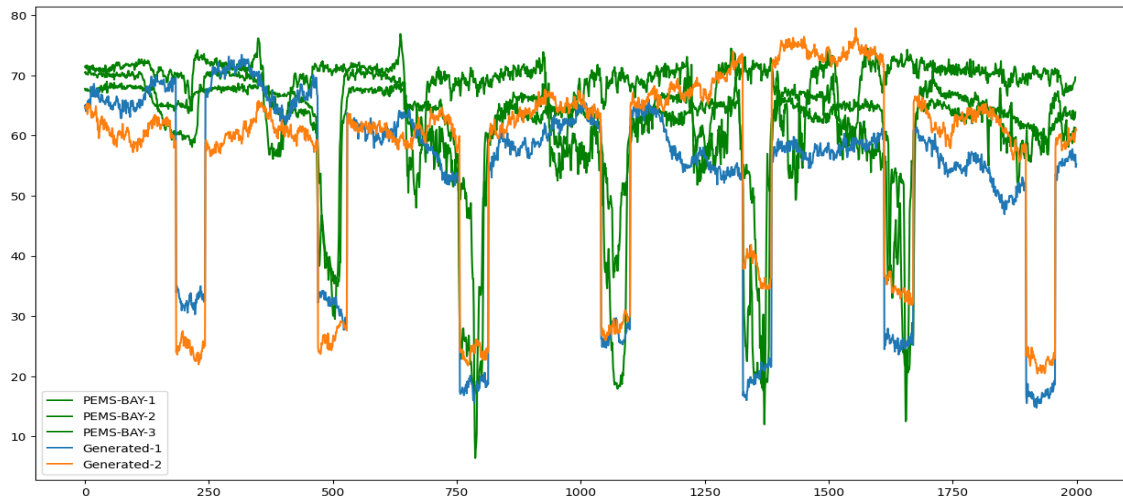


Рис. 9: Датасет PEMS-BAY и синтетические временные ряды

Заключение

В результате работы был реализован генератор временных рядов, поддерживающий различные методы сэмплирования исходных данных и генерации параметров процессов. Были выполнены следующие задачи:

1. Проведен обзор существующих решений для генерации временных рядов.
2. Спроектирована модель генератора временных рядов, которая включает в себя:
 - (a) Набор стохастических процессов, порождающих временные ряды.
 - (b) Метод построения комплексных моделей для генерации временных рядов.
 - (c) Методы создания исходных данных: сэмплирование точек на поверхности второго порядка и генерация случайного графа.
 - (d) Методы генерации параметров временных рядов: метод агрегации и метод параметризации.
3. Выполнена программная реализация модели.

4. Проведена проверка результатов работы генератора.

Код реализации:

<https://github.com/stil-mmo/time-series-generator>

Список литературы

- [1] Abolghasemi Mahdi, Tarr Garth, Bergmeir Christoph. Machine learning applications in hierarchical time series forecasting: Investigating the impact of promotions // [International Journal of Forecasting](#). — 2024. — Vol. 40, no. 2. — P. 597–615.
- [2] Brockwell Peter J., Davis Richard A. Introduction // Introduction to time series and forecasting. — Springer, 2016.
- [3] Controllable Data Generation by Deep Learning: A Review / Shiyu Wang, Yuanqi Du, Xiaojie Guo et al. // [ACM Comput. Surv.](#) — 2024. — mar. — URL: <https://doi.org/10.1145/3648609>.
- [4] Correlated synthetic time series generation for energy system simulations using Fourier and ARMA signal processing / Paul W. Talbot, Cristian Rabiti, Andrea Alfonsi et al. // [International Journal of Energy Research](#). — 2020. — Vol. 44, no. 10. — P. 8144–8155.
- [5] Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting / Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu // International Conference on Learning Representations (ICLR '18). — 2018.
- [6] Diffusion convolutional recurrent neural network: Data-driven traffic forecasting / Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu // arXiv preprint arXiv:1707.01926. — 2017.
- [7] Generation of Realistic Synthetic Financial Time-series / Mihai Dogariu, Liviu-Daniel Ștefan, Bogdan Andrei Boteanu et al. // [ACM Transactions on Multimedia Computing, Communications, and Applications](#). — 2022. — Vol. 18, no. 4. — P. 1–27.
- [8] Graph Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting / Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu // ArXiv. — 2017. — Vol. abs/1707.01926. — URL: <https://api.semanticscholar.org/CorpusID:195346050>.

- [9] Hyndman Rob J., Athanasopoulos George. Exponential smoothing // Forecasting: Principles and practice. — 3rd edition. — OTexts, 2021.
- [10] Kang Yanfei, Hyndman Rob J, Li Feng. GRATIS: GeneRAtIng TIme Series with diverse and controllable characteristics. — 2019. — arXiv:1903.02787.
- [11] Makridakis Spyros, Spiliotis Evangelos, Assimakopoulos Vassilios. The M4 Competition: 100,000 time series and 61 forecasting methods // [International Journal of Forecasting](#). — 2020. — Vol. 36, no. 1. — P. 54–74. — M4 Competition.
- [12] Makridakis Spyros, Spiliotis Evangelos, Assimakopoulos Vassilios. The M5 competition: Background, organization, and implementation // [International Journal of Forecasting](#). — 2022. — Vol. 38, no. 4. — P. 1325–1336. — Special Issue: M5 competition.
- [13] NetworkX. — URL: <https://networkx.org/documentation/stable/index.html> (дата обращения: 4 июня 2024 г.).
- [14] NumPy. — URL: <https://numpy.org/doc/stable/> (дата обращения: 18 декабря 2023 г.).
- [15] Python. — URL: <https://docs.python.org/3/tutorial/index.html> (дата обращения: 18 декабря 2023 г.).
- [16] Time Series Dataset Survey for Forecasting with Deep Learning / Yan-nik Hahn, Tristan Langer, Richard Meyes, Tobias Meisen // [Forecasting](#). — 2023. — Vol. 5, no. 1. — P. 315–335. — URL: <https://www.mdpi.com/2571-9394/5/1/17>.
- [17] Transformers in time series: A survey / Qingsong Wen, Tian Zhou, Chaoli Zhang et al. // arXiv preprint arXiv:2202.07125. — 2022.
- [18] matplotlib. — URL: <https://matplotlib.org/stable/> (дата обращения: 18 декабря 2023 г.).
- [19] mockseries. — <https://github.com/cyrilou242/mockseries>.

- [20] scikit-learn. — URL: <https://scikit-learn.org/stable/index.html> (дата обращения: 19 декабря 2023 г.).
- [21] A survey of deep learning and foundation models for time series forecasting / John A Miller, Mohammed Aldosari, Farah Saeed et al. // arXiv preprint arXiv:2401.13912. — 2024.
- [22] timeseries-generator. — <https://github.com/Nike-Inc/timeseries-generator>.
- [23] Воронцов К.В. Машинное обучение. Прогнозирование временных рядов. К.В. Воронцов, Школа анализа данных, Яндекс. — 2020. — URL: <https://youtu.be/Rmh6b96u6UU?si=o3I20WlIP5EKW2kw> (дата обращения: 19 декабря 2023 г.).
- [24] Документация mockseries. — URL: <https://mockseries.catheu.tech/docs/tutorials/interaction-scalar-operations> (дата обращения: 18 декабря 2023 г.).

Приложения

Приложение 1

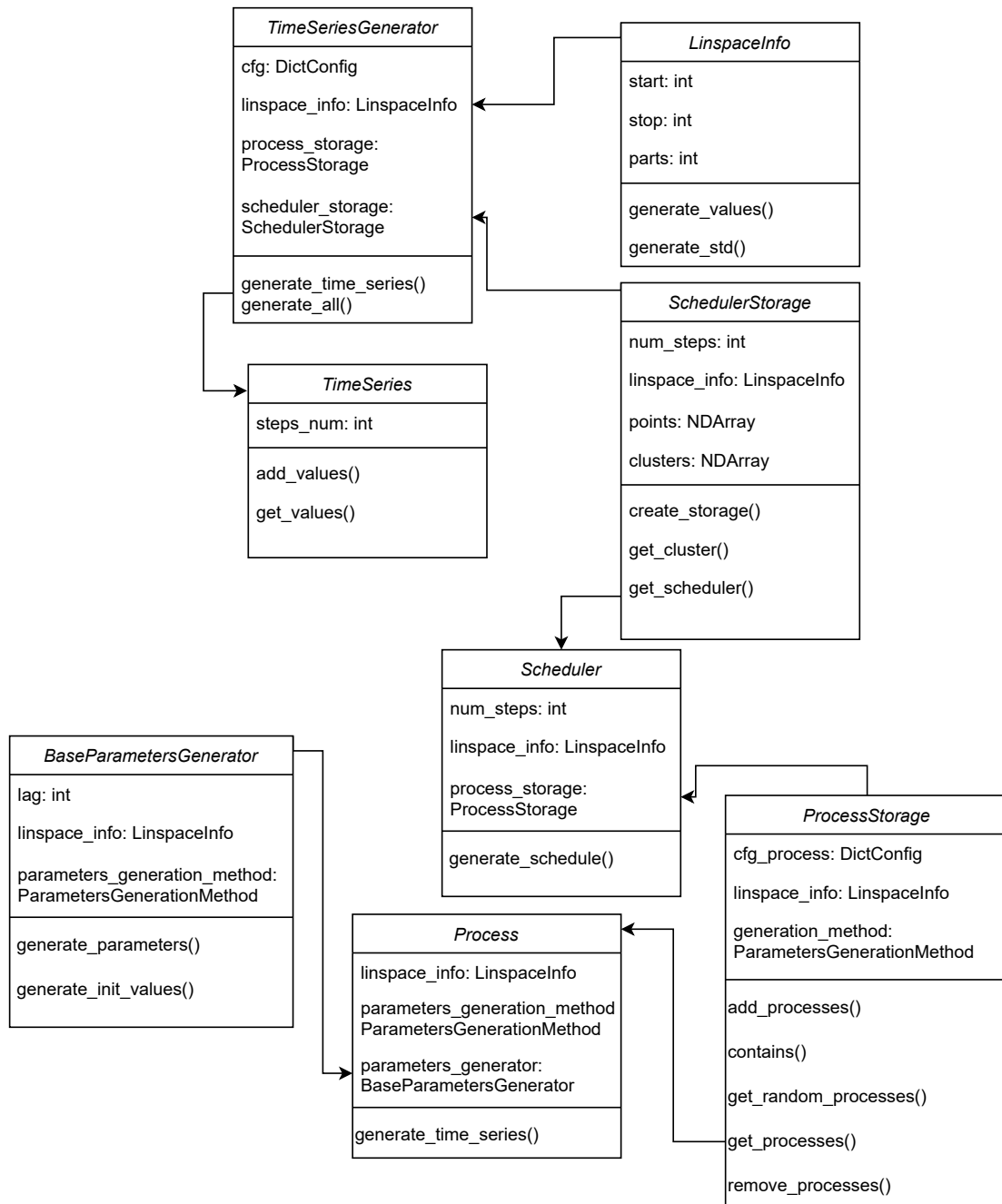


Рис. 10: UML-диаграмма архитектуры генератора временных рядов