Московский физико-технический институт
(государственный университет)
Кафедра Интеллектуальные системы

Работа допущена к защите
зав. кафедрой

_____ Рудаков К.В.

«____» _____ 2014 г.

Выпускная квалификационная работа на степень бакалавра

Тема: **Обнаружение аномалий в дискретных временных рядах**

Направление: 010900 – Прикладные математика и физика

Выполнил студент гр. 074 _____ Неклюдов К.О.

Научный руководитель,
д. ф.-м. н. _____ Воронцов К.В.

Москва – 2014

Оглавление

1. Введение

В данной работе рассматривается прикладная задача обнаружения аномалий в полётных данных. Эта задача относится к более широкому классу задач обнаружения аномалий во временных рядах. Главной особенностью рассматриваемой задачи является отсутствие экспертной разметки исходных данных. Также важной особенностью задачи обнаружения аномалий является отсутствие формального определения аномальности и аномалий. Обычно эти понятия формализуются в ходе исследования в зависимости от выбранного метода. В этой работе мы поступим аналогичным образом, считая формализацию понятия аномальности одной из целей исследования.

Все результаты, полученные в ходе вычислительного эксперимента, получены на реальных данных. Исходными данными задачи является множество полётов, где каждый полёт описан совокупностью показаний датчиков. Показания каждого датчика записывались через равные промежутки времени на протяжении всего полёта, таким образом показания отдельного датчика можно рассматривать как временной ряд. Соответственно, каждый полёт описан набором временных рядов одинаковой длины.

Целью работы является построение алгоритма, который выделяет аномальные полёты на всём множестве полётов и локализует аномалии в каждом полёте.

1.1. Постановка задачи

В работе исследуется задача обнаружения аномалий в дискретных временных рядах. Исследование, описываемое в работе, ориентировано на решение прикладной задачи поиска аномалий в полётных данных.

Опишем исследуемое множество и сразу введём обозначения, которые используются в дальнейшем. Пусть заданы показания датчиков X^i_{jt}, где i – индекс полёта, $j \in J$ – индекс датчика, t – момент времени. Опуская некоторые из индексов, будем пользоваться следующими обозначениями: X^i – полёт (векторный временной ряд размерности $|J|$) длины T_i, X^i_j – показания j-го датчика на протяжении i-го полёта (скалярный временной ряд). Будем полагать, что множество датчиков делится на непрерывные J_c и дискретные J_d.

Ввиду особенностей задачи и её прикладного характера невозможно задать строгий критерий качества и оценить качество алгоритма численно. В большинстве работ в этой области качество алгоритма оценивалось с помощью экспертных оценок, то есть результаты алгоритма подвергались дальнейшему анализу со стороны экспертов, которые выносили заключение о результатах работы алгоритма. В данной работе экспертные оценки не используются. Исходя из этого были поставлены следующие **цели исследования:** найти аномальные полёты и формализовать понятие аномальности при отсутствии информации об аномалиях от экспертов. В работе строится алгоритм, который каждому полёту X^i ставит в соответствие "рейтинг аномальности" $A(X_i)$. Таким образом полёты ранжируются по аномальности.

Исходя из специфики исходных данных были вынесены следующие предположения.

1. Каждый полёт состоит из набора последовательных участков однородности – фаз.
2. Каждая фаза может быть разбита на более мелкие участки однородности – сегменты.
3. Аномалия – маловероятное событие, как внутри полёта, так и на множестве полётов.

1.2. Предыдущие работы

В большинстве работ задача обнаружения аномалий решается методами анализа дискретных временных рядов. Пусть Σ — конечный алфавит, $S_i = s_{i1}, \ldots, s_{iT_i}$ — дискретная последовательность, где $s_{it} \in \Sigma$, $|S_i| = T_i$ — длина последовательности. Всю совокупность данных будем обозначать через $\mathbf{S} = \{S_i\}_{i=1}^n$.

Методы анализа, используемые в предыдущих работах, разделяются на три типа. В обзоре работ приведены основные идеи каждого из трёх методов анализа.

Методы, использующие попарные расстояния между объектами.

1. На множестве $\mathbf{S}$ задаётся расстояние. Расстояние может задаваться многими способами: посимвольное сравнение рядов [1]; длина наибольшей общей подпоследовательности [2]; расстояние основанное на Колмогоровской сложности [3].

2. Вычисляется матрица попарных расстояний между объектами S_i.

3. Объекты кластеризуются. При кластеризации могут применяться различные методы: k медоидов использовался в [2]; одноклассовый метод опорных векторов использовался в [4]; метод k ближайших соседей использовался в [5].

4. Аномальность объекта определяется расстоянием до ближайшей группы.

Оконные методы

Вместо построения оценки аномальности последовательности S_i целиком можно использовать окно ω_{it} длины k, движущееся по времени: $\omega_{it} = \omega_{it}[1]\omega_{it}[2]\ldots\omega_{it}[k]$. Аномальность оценивается отдельно для каждого окна, затем оценки аномальности окон $A(\omega_{i1}), A(\omega_{i2}), \ldots, A(\omega_{iT_i})$ комбинируются в оценку аномальности ряда $A(S_i)$.

Оконные методы различаются определением аномальности окна и аномальности ряда [6–8].

Марковские методы

Вероятность последовательности факторизуется по цепному правилу:

$$P(S_i) = \prod_{i=1}^{T} P(s_{it}|s_{i1}s_{i2}\ldots s_{i(t-1)}).$$

Предполагается, что у процесса короткая память:

$$P(s_{it}|s_{i1}s_{i2}\ldots s_{i(t-1)}) = P(s_{it}|s_{i(t-k+1)}\ldots s_{i(t-1)}), k > 1.$$

Аномальность – величина обратная вероятности: $A(S_i) = \frac{1}{P(S_i)}$.

Марковские методы нахождения аномалий различаются способами подсчёта вероятности [9, 10].

Аналогичные методы существуют и для многомерных временных рядов (как дискретных, так и непрерывных). Отдельно стоит отметить, что во всех предыдущих работах использовалась экспертная оценка или рассматривалась задача обучения с учителем. В задаче обучения с учителем алгоритм использует обучающую выборку с заданными на ней ответами (например, какие полёты являются аномальными). Для оценки качества такого алгоритма используется контрольная выборка с заданными на ней ответами.

Основные отличия данной работы от уже существующих.

- В работе рассматривается задача выделения фаз полёта. Согласно авиационным стандартам, весь полёт может быть разбит на фазы (например, взлёт, набор высоты, посадка и т. д.).

- В работе не используются экспертные оценки. В исходных данных нет никакой дополнительной информации об аномальности полётов. Также не проводится анализ результатов алгоритма с помощью экспертов.

- Разнотипность данных. Каждый полёт представлен набором как дискретных временных рядов, так и непрерывных.

- Аномалии локализуются внутри полёта. В результате работы алгоритма каждый полёт разбивается на участки однородности — сегменты. Считается, что аномальными могут быть отдельные сегменты или последовательность сегментов.

2. Метод обнаружения аномалий

2.1. Схема алгоритма

Приведённый в работе алгоритм относится к методам, использующим попарные расстояния между объектами. Схема работы алгоритма приводится на рисунке 1. Алгоритм состоит из нескольких последовательных этапов:

1. **Выделение фаз полёта.** На этом этапе каждый полёт разбивается на участки однородности, причём каждый участок однородности несёт определённый физический смысл. После разбиения на фазы дальнейший анализ производится для каждой фазы в отдельности.
2. **Приведение непрерывных датчиков к стационарным временным рядам.** Результатом этого этапа являются временные ряды разностей ΔX_j, где $j \in J_c$.
3. **Сглаживание непрерывных временных рядов.** На этом этапе непрерывные временные ряды сглаживаются для уменьшения уровня зашумлённости данных.
4. **Дискретезация непрерывных временных рядов.** Каждому значению X^i_{jt} для непрерывных датчиков $j \in J_c$ сопоставляется символ из заданного алфавита Σ.
5. **Сегментация фаз.** На этом этапе каждая фаза разбивается на более мелкие участки однородности – сегменты.
6. **Кластеризация сегментов.** Полученные на предыдущем этапе сегменты кластеризуются.
7. **Ранжирование по аномальности.** Получение финальных результатов в виде ранжированного по аномальности списка полётов.

Далее подробно рассматривается задача каждого из этапов.

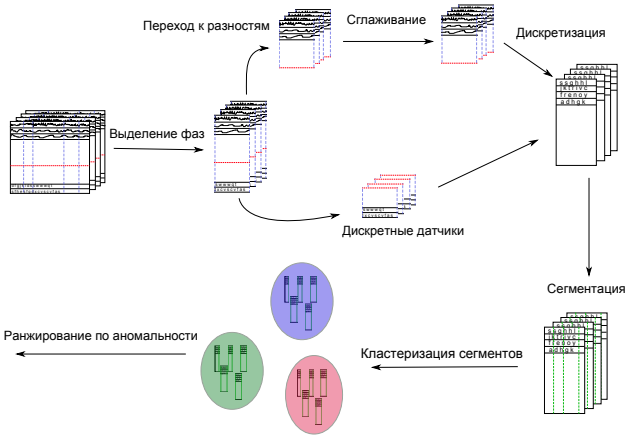


Рис. 1. Схема алгоритма

2.2. Выделение фаз полёта

Как упомяналось выше, полёт может быть разбит на участки однородности – фазы, причём каждая фаза несёт определённый смысл. Выделяют следующие фазы:

- стоянка;
- буксировка от аэропорта;
- руление до взлетной полосы;
- руление до взлета;
- взлет;
- набор круизной высоты;
- круиз;
- снижение;
- маневрирование;
- приближение;
- посадка;
- руление до аэропорта.

Описание каждой фазы можно найти в [11].

В исходных данных содержится разбиение на фазы, но это разбиение не соответствует участкам однородности в показаниях датчиков, что затрудняет дальнейший анализ. Выделим фазы, соответствующие участкам однородности временных рядов, используя следующие датчики:

- скорость самолёта;
- высота;
- угол тангажа;
- угол крена;
- угол атаки;
- расход топлива.

Обозначим множество индексов этих датчиков J_p.

Задача выделения фаз полёта.

Вход: полёт $\left\{X_{jt}^i\right\}_{t=1}^{T_i}$, для заданного множества датчиков $j \in J_p$.

Выход: $\{t_1^i, t_2^i, \ldots, t_{12}^i\}$ — моменты времени, соответствующие началам фаз.

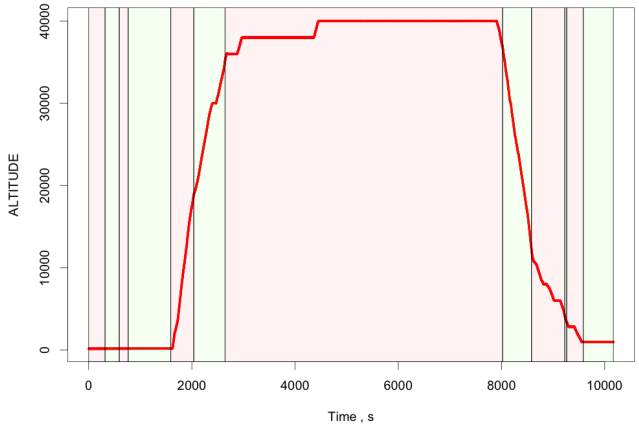
Для выделения фаз используется иерархическая кластеризация с ограничениями [12]. В качестве объектов для кластеризации рассматриваются векторы – показания датчиков $X_t^i = (X_{jt}^i : j \in J_p)$ в данном полёте i для всех моментов времени t. Предварительно все датчики

стандартизуются (из показаний датчика необходимо вычесть мат. ожидание и поделить на дисперсию) для снятия влияния абсолютных значений при кластеризации. Расстояние между объектами евклидово. Расстояние между кластерами – расстояние Уорда. Ограничение, накладываемое на кластеризацию, состоит в том, что объединяться могут только соседние по времени кластеры.

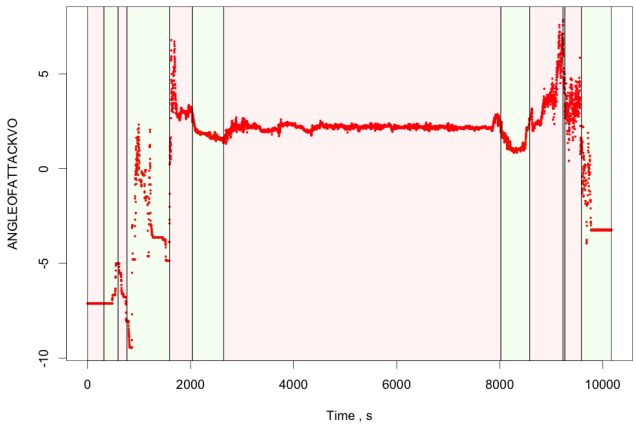
$$R(W,S) = \frac{|U|+|S|}{|S|+|W|}R(U,S) + \frac{|V|+|S|}{|S|+|W|}R(V,S) - \frac{|S|}{|S|+|W|}R(U,V).$$

Здесь $W = U \cup V$, а U,V — объединяемые кластеры.

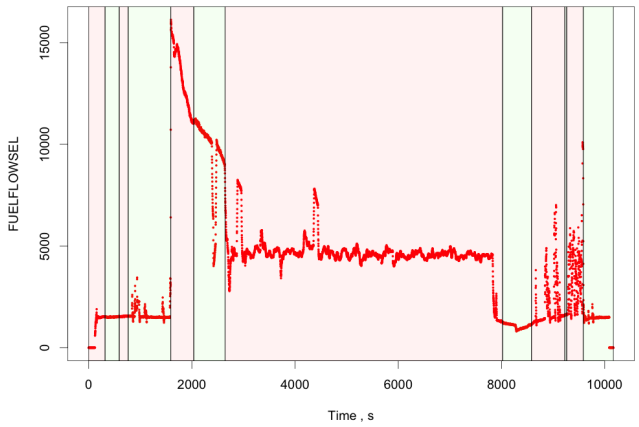
На рис. 2 приведены показания различных датчиков, наложенные на полученные разбиения по фазам.



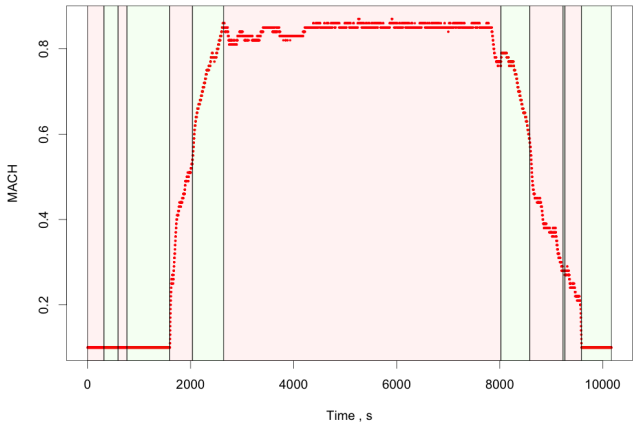
Высота



Угол атаки



Расход топлива



Скорость в махах

Рис. 2. Показания датчиков

Из графиков видно, что разбиение на фазы соответствует участкам однородности в показаниях датчиков.

На Рис. 3 полученное разбиение сравнивается с разбиением на фазы, предоставленном в исходных данных.

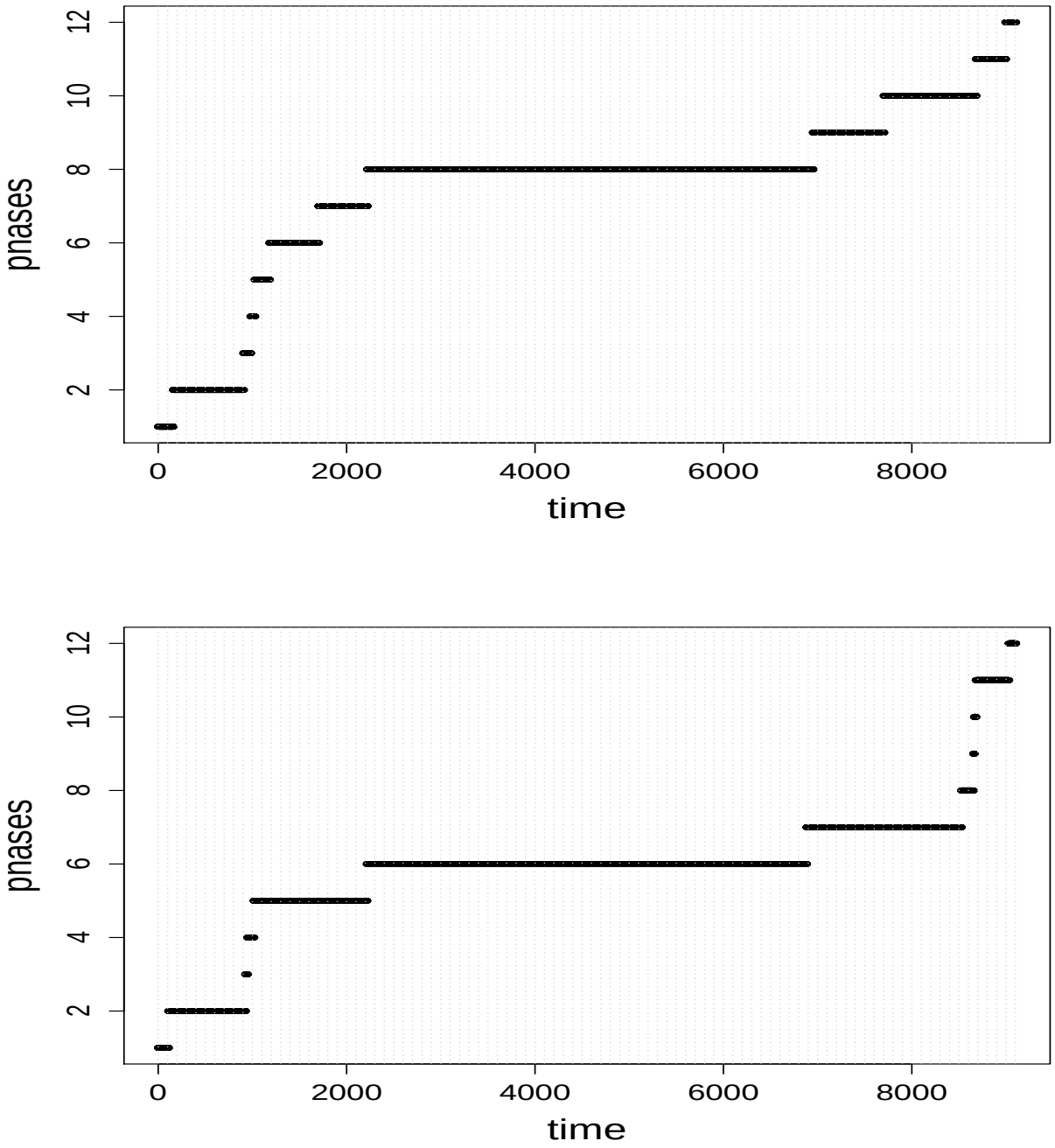


Рис. 3. Рассчитаные фазы (вверху)

Фазы в исходных данных (внизу)

Из графиков видно, что полученное разбиение согласуется с разбиением в данных.

После разбиения каждого полёта на фазы дальнейший анализ производится внутри каждой фазы в отдельности. Все дальнейшие результаты иллюстрируются на примере круизной фазы полёта.

2.3. Приведение непрерывных датчиков к стационарным временным рядам

Для дальнейшего анализа необходимо привести непрерывные временные ряды, соответствующие показаниям непрерывных датчиков, к стационарным временным рядам.

Задача приведения непрерывных датчиков к стационарным временным рядам.

Вход: X_j – показания j-го датчика во всех полётах внутри выбранной фазы.

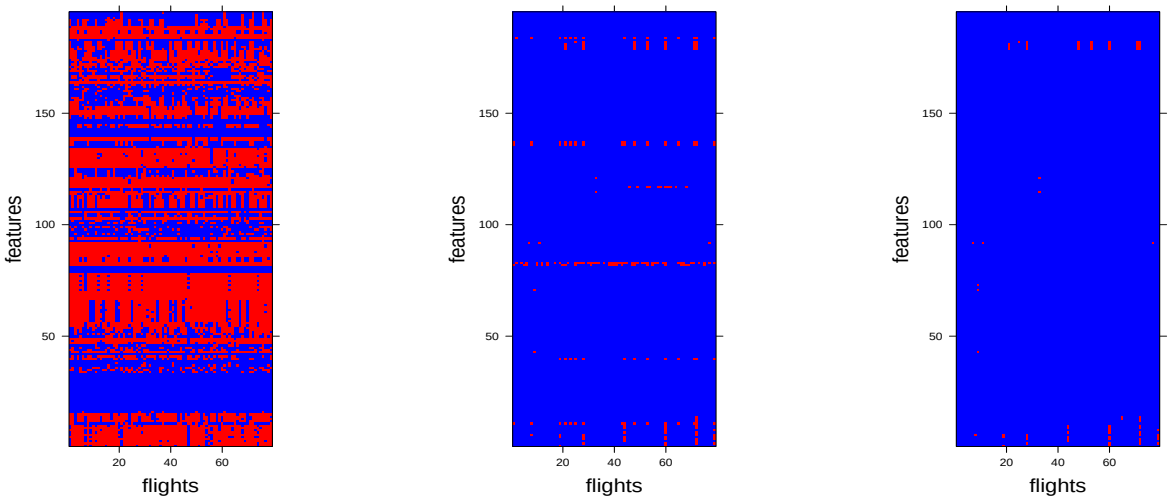
Выход: стационарные временные ряды разностей ΔX_j.

Для отдельного датчика переход к разностям будет $\Delta X_{jt} = X_{j(t+1)} - X_{jt}$.

Чтобы снять влияние абсолютных значений датчиков, переходим к разностям во всех датчиках. Но после этого перехода могут остаться нестационарные ряды, в которых необходимо сделать переход к разностям ещё раз. Таким образом, необходим инструмент проверки стационарности временного ряда. В качестве такого инструмента предлагается использовать критерий KPSS [13], который проверяет нулевую гипотезу о стационарности временного ряда против альтернативы, что временной ряд имеет линейный тренд.

Для конкретного датчика $j \in J_c$ рассматривался набор временных рядов $\{X_j^i\}_{i=1}^N$ для всех полётов. Решение о переходе к попарным разностям принималось исходя из результатов критерия KPSS для всего набора полётов. Точнее, если число полётов, в которых временной ряд для данного датчика не стационарен, превышает заданный порог, тогда принимается решение о переходе к попарным разностям для этого датчика во всех полётах.

Ниже приведены результаты критерия KPSS, где красным отмечены клетки, для которых гипотеза стационарности была отвергнута.



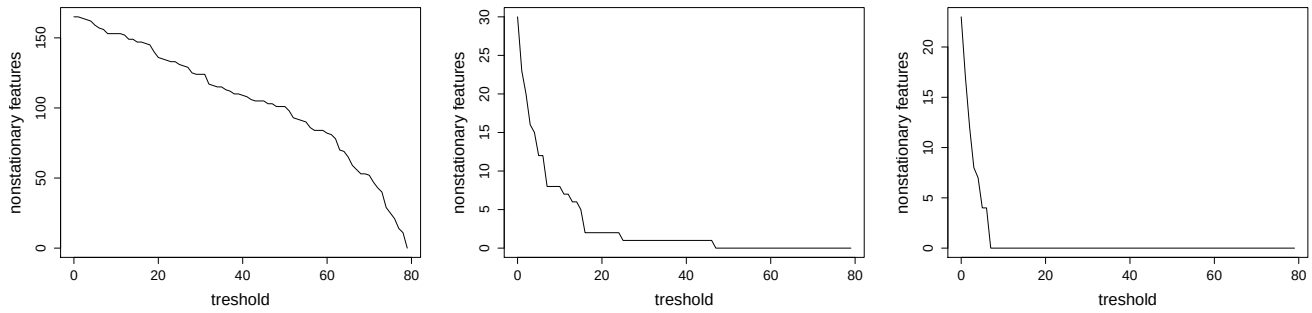
Исходные ряды | первые разности | вторые разности

Рис. 4. Результаты критерия KPSS

Из графиков видно, что после перехода к попарным разностям нулевая гипотеза в кри-

терии KPSS не может быть отвергнута в большинстве полётов.

Рисунок 5 показывает, для какого числа датчиков необходим переход к разностям при различных значениях порога.



До перехода к разностям Один переход Два перехода

Рис. 5. Выбор порога

Нами был выбран порог в 10 полётов. По рисунку 5 видно, что для такого порога достаточно второго перехода к разностям.

2.4. Сглаживание непрерывных временных рядов

После взятие попарных разностей уровень шума в данных резко возрастает, что сильно влияет на последующие этапы алгоритма и, как следствие, на результат работы. Для снижения уровня шума временные ряды сглаживаются.

Пример сглаживания:

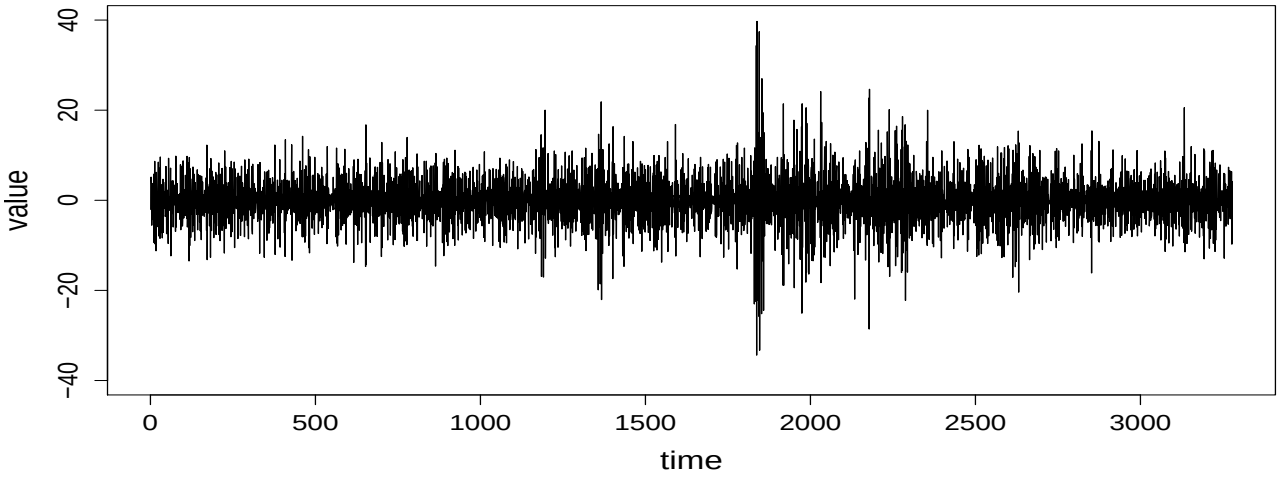


Рис. 6. Исходный ряд

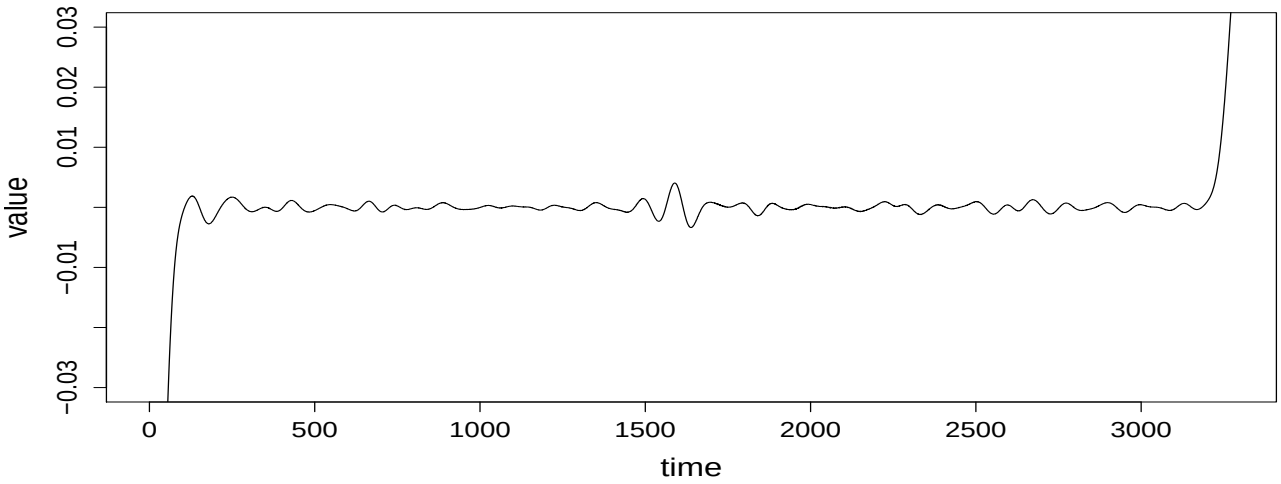


Рис. 7. Сглаженный ряд

Краевые эффекты на сглаженной функции объясняются методом сглаживания.

К временному ряду применялось ядерное сглаживание, что, по сути, является свёрткой двух функций:

$$\widehat{f}(t) = \sum_{i=-N}^{N} f(t+i)K(i)$$

Здесь $2N$ — размер окна, f — исходная функция, K — гауссовское ядро.

$$K(u) = \frac{1}{\sqrt{2\pi}} exp\left(\frac{-u^2}{2}\right)$$

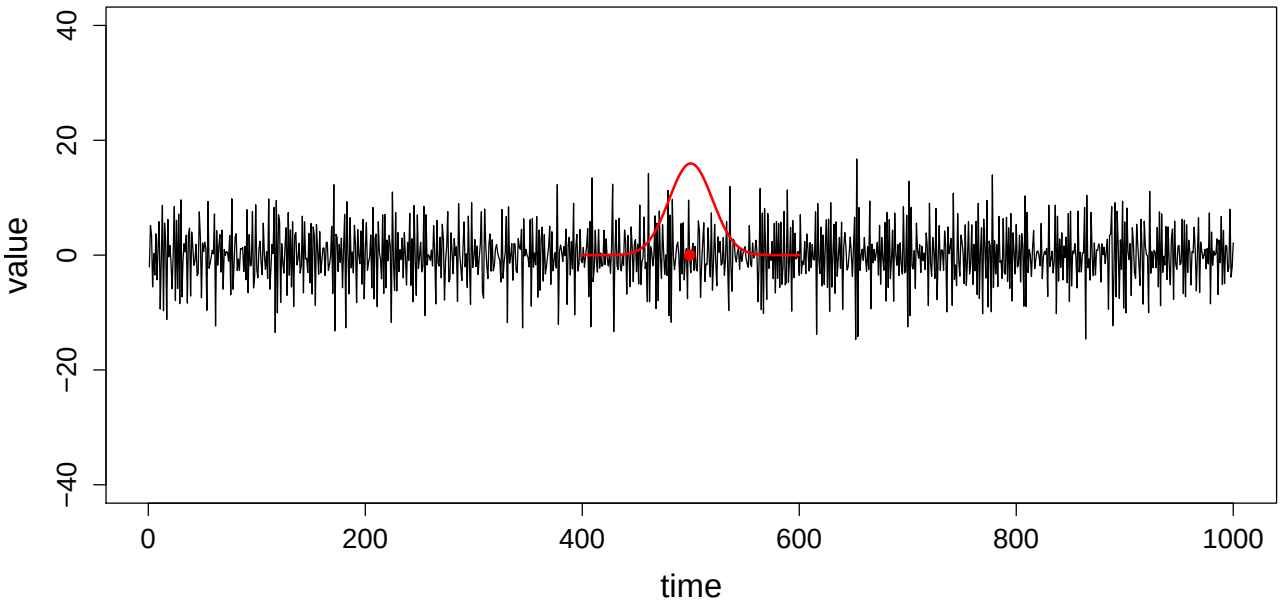


Рис. 8. Гауссовское ядро

2.5. Дискретезация непрерывных временных рядов

На этапе дискретизации каждому значению X^i_{jt}, где $j \in J_c$, ставится в соответствие символ из алфавита Σ.

Вход: $\{X^i_{jt}\}$, где $j \in J_c$,

$t \in [t^i_k; t^i_{k+1}]$, где $[t^i_k; t^i_{k+1}]$ – k-ая фаза i-го полёта.

Выход: дискретизованные значения $\{X_{jt}\}$.

2.6. Сегментация фаз

На этапе сегментации каждая фаза разбивается на участки однородности – сегменты.

Вход: k-ая фаза полёта X^i – $|J|$ временных рядов длины $t^i_{k+1} - t^i_k$.

Выход: $S_i = \{s_l\}_{l=1}^{L_i}$ – набор сегментов в фазе полёта X^i

2.7. Кластеризация сегментов

На этом этапе все сегменты кластеризуются.

Вход: $\{s_l\}_{l=1}^{L}$ – множество всех сегментов. $L = \sum_{i=1}^{N} L_i$

Выход: метки кластеров для каждого сегмента s_l

Перечисленные этапы алгоритма подробно описаны в работе [15]. На этих этапах алгоритма каждому сегменту полёта ставится в соответствие символ из алфавита, заданного на всём множестве полётов. Результатом работы является представление каждого полёта как одномерного дискретного ряда.

Исследуемые данные содержали записи полётов двух самолётов одного типа. Но при кластеризации сегментов на два кластера данные разделяются — сегменты полёта одного самолёта попадают в один кластер.

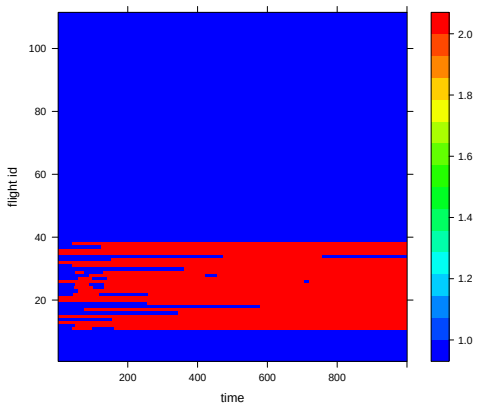


Рис. 9. Разделение данных

Дальнейший анализ производился для полётов одного и того же самолёта.

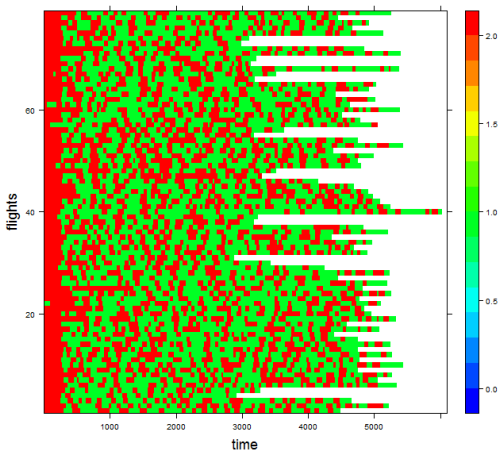
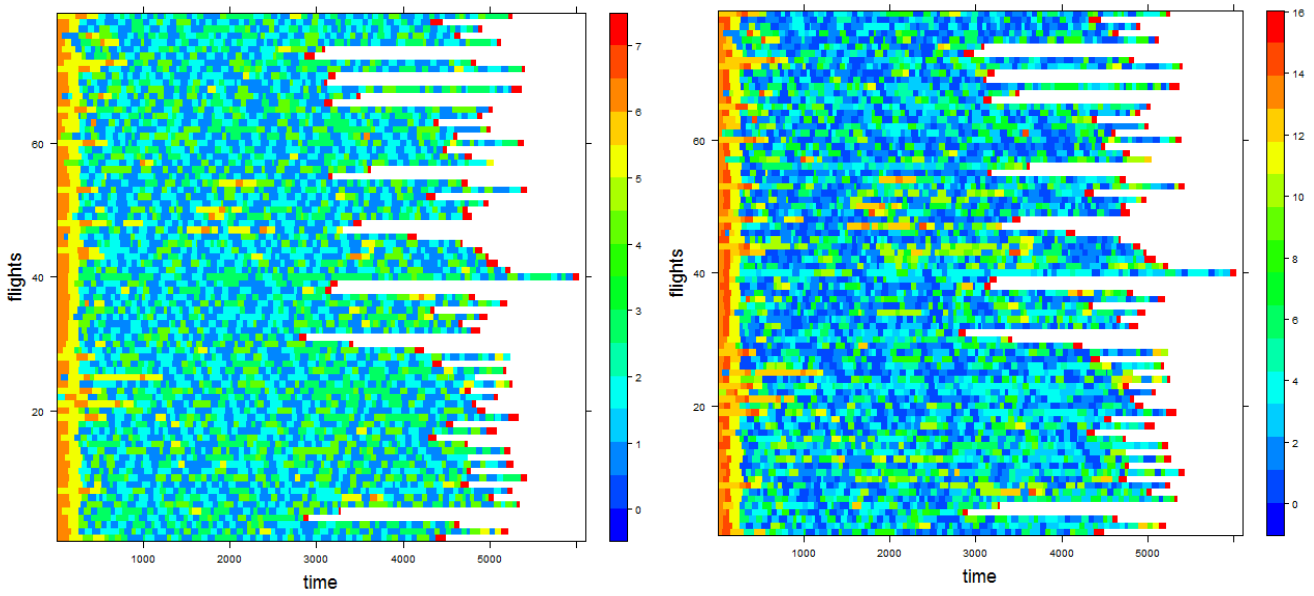


Рис. 10. Полёты не разделяются

Как видно из графика 10, аналогичная кластеризация для одного самолёта не даёт таких же результатов. Приведём результаты кластеризации при других значениях обшего числа кластеров.



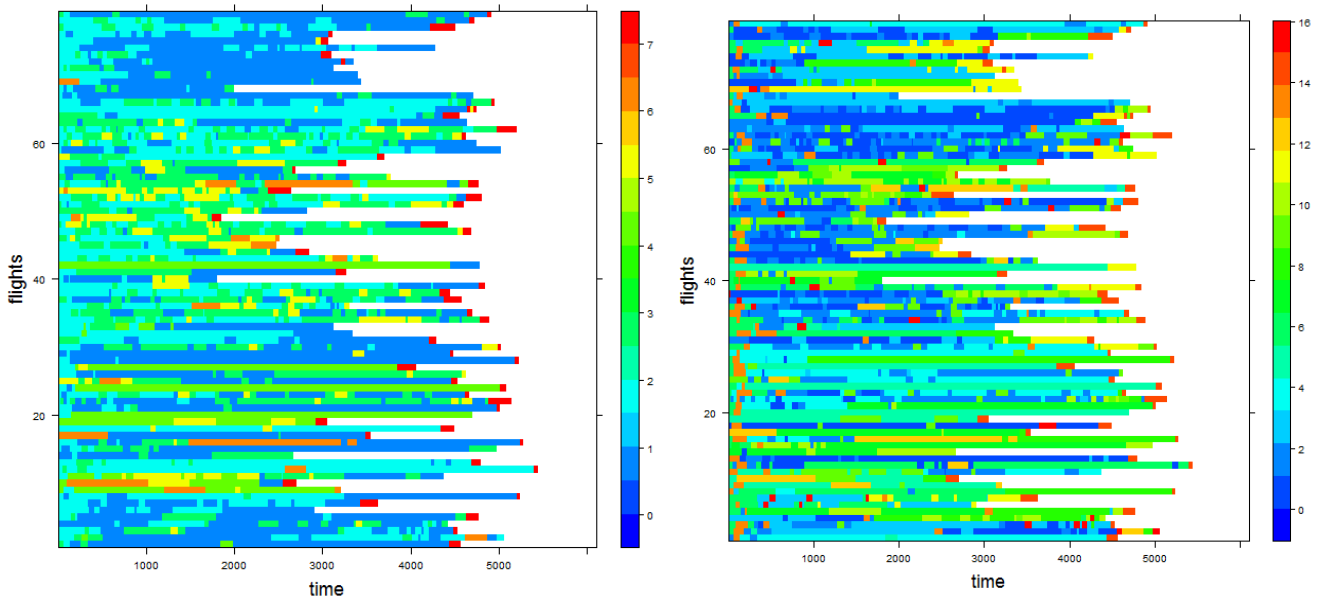
7 кластеров

15 кластеров

Рис. 11. Кластеризация

Характерным на этих графиках является то, что полёты кластеризуются одинаково: каждая круизная фаза содержит вначале сегменты одного и того же кластера, затем следуют середины круизных фаз, которые представляют из себя набор близких кластеров, и концы каждой круизной фазы также лежат в одном кластеры. Таким образом, круизные фазы сегментировались на “начало”, “середину” и “конец”.

Также представим результаты для несглаженных временных рядов (рис. 12). В таком случае похожей кластеризации не наблюдается.



7 кластеров

15 кластеров

Рис. 12. Кластеризация

Круизные фазы не разбиваются на “начало”, “середину” и “конец”, т.е. круизные фазы перестали быть “похожи”, следовательно, дальнейший анализ не имеет смысла.

2.8. Ранжирование по аномальности

Ранжирование по аномальности является финальным этапом алгоритма. На выходе мы получаем ранжированный по аномальности список полётов.

Вход: $\{\tilde{X}_i\}_{i=1}^{N}$ — набор фаз полётов, представленных в виде одномерных дискретных рядов.

Выход: список полётов, отранжированных по аномальности.

На этом этапе каждая круизная фаза представляется как дискретный временной ряд. Элементом такого ряда является номер кластера, к которому был отнесён очередной сегмент. При анализе полётов необходимо учитывать порядок следования элементов, а также значения самих элементов. Исходя из этих предпосылок, для сравнения полётов была выбрана nLCS метрика [2]:

$$nLCS(X,Y) = 1 - \frac{LCS(X,Y)}{\max(l_X, l_Y)}$$

Здесь $LCS(X,Y)$ – длина наибольшей общей подпоследовательности в строчках X и Y; l_X, l_Y, соответственно, длины строк X и Y.

Введя метрику между круизными фазами, мы получаем матрицу попарных расстояний. Среди круизных фаз выберем фазу, которая является центром группы в этой метрике. Это задача поиска центра кластера, когда кластер всего один. Решение такой задачи может быть найдено итеративным алгоритмом, например, метод k медоидов [14]. В нашем случае мы имеем 79 круизных фаз, и центр такой группы может быть найден полным перебором. Элемент, который соответствует центру группы, объявляется эталоном, и, соответственно, все полёты ранжируются по аномальности в зависимости от расстояния до эталона.

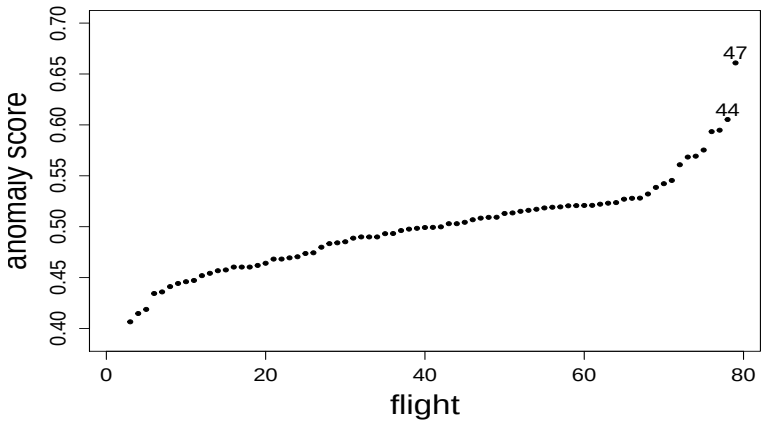


Рис. 13. Пример ранжирования (15 кластеров)

Также исследуем наш алгоритм на устойчивость. Для этого проведём несколько кластеризаций сегментов с различным числом кластеров и посмотрим на 5 самых аномальных полётов.

число кластеров =	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
топ 1	47	4	40	40	40	47	23	30	30	30	47	47	47	47	47	47
топ 2	19	40	4	4	4	23	47	23	23	23	62	23	44	44	44	44
топ 3	40	26	23	23	23	56	30	47	65	62	30	65	65	30	30	65
топ 4	68	5	26	26	5	65	56	56	62	65	23	13	30	65	65	43
топ 5	48	39	62	62	26	5	62	62	47	47	43	40	43	43	43	30

Как видим, в таблице наблюдаются номера полётов, которые входят в 5 самых аномальных при различном числе кластеров, что говорит об устойчивости алгоритма, относительно этого параметра.

3. Заключение

В работе описан алгоритм поиска аномалий в полётных данных без использования разметки данных или экспертной оценки. Также приведены результаты для реальных данных. Данный алгоритм обладает рядом преимуществ:

1. основной результат выдаётся в виде ранжированного списка, как в поисковых системах;
2. аномалии локализуются внутри фаз и сегментов;
3. не требуется экспертная разметка;
4. метод можно использовать как инструмент разведочного анализа данных.

Список литературы

1. R. R. Sokal, "A statistical method for evaluating systematic relationships," *Univ Kans Sci Bull*, vol. 38, pp. 1409–1438, 1958.
2. S. Budalakoti, A. N. Srivastava, and M. E. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 101–113, 2009.
3. E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S.-H. Lee, and J. Handley, "Compression-based data mining of sequential data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 99–129, 2007.
4. S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 47–56, ACM, 2010.
5. V. Chandola, V. Mithal, and V. Kumar, "Comparative evaluation of anomaly detection techniques for sequence data," in *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pp. 743–748, 2008.
6. C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, pp. 133–145, IEEE, 1999.
7. S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of computer security*, vol. 6, no. 3, pp. 151–180, 1998.
8. T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 3, pp. 295–331, 1999.
9. C. C. Michael and A. Ghosh, "Two state-based approaches to program-based anomaly detection," in *Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference*, pp. 21–30, IEEE, 2000.
10. C. Marceau, "Characterizing the behavior of a program using multiple-length n-grams," in *Proceedings of the 2000 workshop on New security paradigms*, pp. 101–110, ACM, 2001.
11. I. C. A. O. Common taxonomy team, "Phase of flight: Defenition and usage notes," http://www.intlaviationstandards.org/Documents/PhaseofFlightDefinitions.pdf.
12. E. C. Grimm, "Coniss: a fortran 77 program for stratigraphically constrained cluster analysis

by the method of incremental sum of squares," *Computers & Geosciences*, vol. 13, no. 1, pp. 13–35, 1987.

13. D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?," *Journal of econometrics*, vol. 54, no. 1, pp. 159–178, 1992.
14. L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program pam)," *Finding groups in data: an introduction to cluster analysis*, pp. 68–125, 1990.
15. Д. Д. Яшков, науч. рук. К. В. Воронцов, Бакалаврская диссертация: "Проблема понижения размерности в задаче поиска аномалий в многомерных временных рядах", МФТИ(ГУ), 2014.