

УДК 519.6+681.51
<http://doi.org/10.18799/29495407/2026/2/124>
Шифр специальности ВАК: 2.3.5



Модификация методов потоковой кластеризации для сегментации временных рядов в реальном времени

Е.А. Кочегурова✉, В.В. Ленский

*Национальный исследовательский Томский политехнический университет,
Российская Федерация, г. Томск*

✉ kocheg@tpu.ru

Аннотация. Рассмотрены задачи сегментации временных рядов и обнаружения разладок в потоковых данных с использованием алгоритмов потоковой кластеризации CluStream и DenStream. Предложен подход к сегментации в реальном времени с расширяющимся окном для накопления отсчетов ряда. Для каждого окна вычисляется набор статистических и структурных признаков, после чего кластеризация выделяет границы сегментов. Проведенные эксперименты на синтетических и реальных данных показали, что алгоритм DenStream точнее обнаруживает переходы между сегментами, а CluStream более устойчив к ложным границам. Разработано программное приложение для исследования потоковой сегментации с графическим интерфейсом и визуализацией. Программная система может применяться в задачах мониторинга и анализа телеметрических данных в реальном времени.

Ключевые слова: временной ряд, сегментация временных рядов, потоковая кластеризация, разладка, CluStream, DenStream, расширяющееся окно, признаковое описание, метрики

Для цитирования: Кочегурова Е.А., Ленский В.В. Модификация методов потоковой кластеризации для сегментации временных рядов в реальном времени. *Известия Томского политехнического университета. Промышленная кибернетика*, 2026, Т. 4, № 2, С. 22–30. <http://doi.org/10.18799/29495407/2026/2/124>

Вклад авторов: В.В. Ленский – представление результатов исследования, Е.А. Кочегурова – теоретическая часть работы.

Использование средств ИИ: авторы заявляют о том, что при написании данной статьи не применялись средства генеративного искусственного интеллекта.

Конфликт интересов: отсутствует.

UDC 519.6+681.51
<http://doi.org/10.18799/29495407/2026/2/124>

Modification of streaming clustering methods for real-time series segmentation

E.A. Kochegurova✉, V.V. Lensky

National Research Tomsk Polytechnic University, Tomsk, Russian Federation

✉ kocheg@tpu.ru

Abstract. This paper examines the problems of time series segmentation and error detection in streaming data using the CluStream and DenStream streaming clustering algorithms. A real-time segmentation approach is proposed using an expanding window for accumulating time series samples. For each window, a set of statistical and structural features is calculated, after which clustering identifies segment boundaries. Experiments conducted on synthetic and real data showed that the DenStream algorithm more accurately detects transitions between segments, while CluStream is more robust to false boundaries. A software application for streaming segmentation research with a graphical interface and visualization was developed. The software system can be used for real-time telemetry data monitoring and analysis.

Keywords: time series, time series segmentation, stream clustering, disorder, CluStream, DenStream, expanding window, feature description, metrics

For citation: Kochegurova E.A., Lensky V.V. Modification of streaming clustering methods for real-time series segmentation. *Bulletin of the Tomsk Polytechnic University. Industrial Cybernetics*, 2026, vol. 4, no. 2, pp. 22–30. <http://doi.org/10.18799/29495407/2026/2/124>

Authors' contribution: V.V. Lensky – presentation of the results of the study; E.A. Kochegurova – the theoretical part of the work.

Use of AI: AI was not used.

Conflict of interest: none.

Введение

Временные ряды (ВР) – это один из способов представления данных о процессах, изменяющихся во времени. Они используются в промышленном мониторинге, телеметрии, анализе сетевого трафика, финансовых системах, обработке медицинских сигналов, системах контроля оборудования и других прикладных областях. Нередко такие данные поступают не в виде заранее сформированного набора наблюдений, а непрерывным потоком. Это предъявляет к методам анализа требование обработки «на лету», т. е. по мере поступления данных и без повторного анализа всей истории наблюдений.

Одной из задач анализа ВР является сегментация, то есть выделение фрагментов ВР, соответствующих различным режимам процесса. Смена режима может проявляться через изменение среднего уровня сигнала, дисперсии, амплитуды колебаний, тренда, частоты или других статистических характеристик. Обнаружение таких переходов позволяет выявлять изменение состояний процесса, фиксировать аномальные или устойчивые режимы работы и использовать результаты анализа в задачах мониторинга и принятия решений [1–4].

Режим реального времени в задаче сегментации накладывает требование критичности к обнаружению начала нового фрагмента, как к изменению состояния процесса. Возможным подходом является обнаружение разладки ВР (ChangePoint) на основе потоковых данных [5–8].

В случае нестационарных ВР универсального подхода к обнаружению разладки в реальном времени не существует. Одним из возможных решений может быть использование методов кластеризации со скользящим окном [4, 9, 10]. Методы потоковой кластеризации позволяют описывать поступающие фрагменты данных через обновляемую кластерную модель. Однако для решения задачи сегментации необходимо дополнительно задать совокупность признаков, определяющих изменение динамики процесса, и таким образом интерпретировать смену кластерной принадлежности как границу между режимами процесса. Задача также усложняется спецификой объектов кластеризации – отсчётами ВР, которые характеризуются двумя ключевыми

параметрами: временными метками и пространственными значениями. Методы кластеризации ВР выделяют группы отсчетов ВР (классы), объединенные близостью как временных меток, так и соответствующих им значений ВР [11].

Потоковая кластеризация

При потоковой кластеризации анализируемые объекты поступают последовательно, а модель кластеров должна обновляться по мере появления новых наблюдений [11–13].

Ключевыми требованиями к методам потоковой кластеризации являются:

- компактное представление уже обработанных данных, так как хранение всего потока обычно невозможно;
- постепенное обновление модели, без полного пересчёта всей истории;
- учет эволюции потока: появление новых групп объектов, изменение плотности существующих групп, наличие шума в данных.

Распространённым способом компактного представления потока данных являются микрокластеры. Они описывают небольшие группы близких объектов с помощью агрегированных характеристик, по которым можно оценивать положение, размер или плотность области.

Для потоковой кластеризации характерна двухуровневая организация обработки [14–16]. На первом уровне выполняется быстрая последовательная обработка новых объектов и обновление компактных структур – микрокластеров. На втором уровне на основе этих микрокластеров может выполняться более крупное обобщение. Например, построение итоговых кластеров или анализ изменений структуры потока. Такая схема позволяет выделить задачу оперативного обновления модели и задачу получения интерпретируемого результата. Именно этот принцип лежит в основе ряда известных алгоритмов потоковой кластеризации, включая CluStream и DenStream

При потоковой кластеризации ВР последовательность его отсчетов заменяют на фрагменты, т. е. участки фиксированной или переменной длины, поскольку отдельный отсчет ВР не отражает режим

или динамику процесса (форму, тренд, шум, амплитуду).

Фрагменты можно формировать через скользящее окно с постоянным шагом или расширять до выполнения определенного условия, т. е. задавать адаптивно. После выделения фрагмента необходимо его описать набором числовых признаков. Это уменьшает размерность задачи и делает сравнение более устойчивым. Таким образом, сопоставление фрагментов происходит по обобщённым свойствам, а не по отдельным отсчетам ВР.

Однако выбор параметров критичен: слишком короткое окно чувствительно к шуму, слишком длинное сглаживает границы. Неудачный набор признаков также снижает качество кластеризации. Таким образом, успешная сегментация требует согласованного подбора окна, признаков и метода кластеризации.

Описание алгоритмов потоковой кластеризации

Для сегментации ВР в реальном времени методы потоковой кластеризации целесообразно выбирать алгоритмы из разных групп. Группы алгоритмов формируются в зависимости от того, как данные сжимаются в реальном времени и как/какой формы формируют финальные кластеры [6].

а) Алгоритм CluStream относится к группе разделимых (partitional) алгоритмов кластеризации сферического типа. DenStream – к группе плотностных алгоритмов.

CluStream – базовый алгоритм потоковой кластеризации, разделяющий процесс на две фазы: онлайн-обновление микрокластеров и офлайн-построение итоговых кластеров. Это позволяет обрабатывать поток в реальном времени и получать интерпретируемую структуру.

Микрокластер, как структурная единица потока, хранит агрегированную статистику параметров группы близких объектов (количество, суммы и суммы квадратов координат, временные метки). Это позволяет вычислять центр и разброс без хранения всех точек. Временная информация используется для оценки актуальности кластера.

На онлайн-стадии каждый новый объект приписывается к ближайшему микрокластеру. Если же расстояние превышает порог, то создаётся новый микрокластер. При ограничении числа микрокластеров в потоковой обработке алгоритм удаляет устаревшие или объединяет близкие. Этим создаётся компактность модели.

На офлайн-стадии из созданных микрокластеров строятся укрупнённые кластеры, отделяющие данные за выбранный период.

Достоинством двухуровневой организации является разделение фазы быстрого обновления от

трудоемкого анализа. Также алгоритм требует малой памяти и довольно устойчив.

б) DenStream – потоковой алгоритм кластеризации, основанный на плотностном подходе. В отличие от центроидных методов (CluStream, StreamKM++) в алгоритме DenStream, кластеры определяются через области близкой плотности в пространстве признаков. Такой подход позволяет выделять кластеры сложной формы, что является большим преимуществом перед центроидными методами с их кластерами правильной формы. Также алгоритм отдельно обрабатывает наблюдения, не относящиеся к устойчивым областям. Иногда такие данные трактуются как шум.

В алгоритме используются два типа микрокластеров: потенциальные (устойчивые плотные области) и выбросы (для временного хранения объектов, не попавших в устойчивые кластеры). Это позволяет не интерпретировать единичное отклонение как новый кластер. Но при этом сохраняется возможность обнаружить новую область при поступлении похожих данных.

Ключевой механизм алгоритма DenStream – это затухание весов. Вклад старых объектов со временем уменьшается, что повышает адаптацию к изменению потока данных. Если область перестаёт пополняться, её вес снижается. Область выбросов без пополнения и подтверждения удаляется. Но в случае подтверждения достаточной плотности эта область переходит в потенциальные микрокластеры. Процедура смены типа группы выброс/потенциальный микрокластер соответствует динамике потоковых данных.

В целом оба алгоритма потоковой кластеризации зависят от выбранных метрик и пространства признаков. Но механизмы выделения кластеров в этих алгоритмах отражают природу сегментации ВР в реальном времени.

Программная реализация

Для проведения экспериментальных исследований разработано программное приложение, предназначенное для реализации алгоритмов потоковой кластеризации и анализа ВР. Программная система обеспечивает последовательную обработку входных данных и автоматическое выделение границ между фрагментами ВР, а также сравнение результатов работы алгоритмов кластеризации и визуализацию процесса сегментации.

Основное назначение приложения – предоставить исследовательский инструмент при работе с потоковыми данными для обнаружения изменений поведения системы в реальном времени. Модульная архитектура программной системы ориентирована на последовательное выполнение всего функционала приложения и включает следующие этапы сегментации:

- генерация или обработка ВР;
- формирование фрагментов (окон) ВР;
- извлечение и нормализация признаков;
- выделение границ сегментов;
- визуализация результатов;
- расчёт метрик качества.

Смысловую нагрузку задачи сегментации ВР выполняет этап «формирование фрагментов (окон) ВР». На этом этапе осуществляется сбор информации о каждом фрагменте ВР: индекс, начало и конец окна, центральная позиция во ВР, значения сигнала внутри окна и сформированный вектор признаков. Это позволяет отделить исходные значения ВР от универсального признакового представления. Алгоритмы кластеризации CluStream и DenStream получают на вход именно признаки, описывающие поведение фрагментов.

Предлагаемая в работе модификация потоковых алгоритмов кластеризации включает понятие «расширяющееся окно». Новые отсчеты добавляются в текущий фрагмент, и оценивается однородность признаков расширенного фрагмента. Если участок остаётся однородным, окно расширяется и далее. При появлении устойчивого различия признаков окно закрывается и фиксируется конец сегмента.

Программно интерфейс потокового алгоритма выполняет предварительную проверку текущего окна без изменения состояния модели. Происходит возврат предполагаемой метки кластера и его совместимости, вычисляется расстояние, радиус и признаки новизны/шума. Далее в модель окончательно добавляются признаки закрытого сегмента, фиксируя решение без повторной классификации.

В текущей реализации для длинных окон используется комбинированное описание: признаки вычисляются как по всему окну, так и по его последней части. Затем признаки смешиваются. Это позволяет учитывать общие свойства сегмента и одновременно быстрее реагировать на недавние изменения. Для признаков уровня и масштаба последние значения имеют больший вес, что ускоряет обнаружение смены режима.

Для числового описания сформированных фрагментов используется набор статистических и динамических признаков. Вектор признаков включает характеристики уровня, рассеяния, тренда, локальной изменчивости и колебательной структуры отсчетов ВР. Статистические признаки описывают центральный уровень значений окна и вариабельность основной массы значений, снижая чувствительность к выбросам.

Динамические признаки более сложны и включают: наклон линейного тренда (направленное изменение), стандартное отклонение первых разностей (локальная изменчивость), долю пересечений среднего уровня (частота смены знака отклонения),

автокорреляцию с лагом 1 (степень плавности), долю высокочастотной составляющей и показатель шероховатости (отношение локальных изменений к общему разбросу).

Совокупность перечисленных признаков формирует компактное векторное представление фрагмента, отражающее не только уровень сигнала, но и его устойчивость, направленную динамику, локальную изменчивость и колебательный характер. Смена режима проявляется как изменение всего признакового вектора, а не отдельных его компонент.

Извлеченные признаки имеют разный масштаб и разные диапазоны значений. Нормализация приводит признаки к единому пространству для сравнения.

Она включает следующие операции: логарифмическое сжатие с сохранением знака для признаков уровня, разброса, тренда и изменчивости; преобразование ограниченных признаков (доли, частоты, шероховатости) из интервала $[0,1]$ в $[-1,1]$; взвешивание признаков согласно их вкладу в разделение режимов.

Нормализация особо важна для потоковых алгоритмов, т. к. в процессе обработки «на лету» признаки должны сравниваться в стабильном пространстве на протяжении всего ВР.

Пользовательский интерфейс приложения представлен на рис. 1 и включает набор вкладок: «Старт», «Метрики», «Наборы данных» и «Логги». Основной экран приложения построен на базе вкладок `ttk.Notebook`, а графики встраиваются в окно приложения с помощью средств `Matplotlib`.

На вкладке «Старт» пользователю предоставляется возможность:

- выбора набора данных и алгоритмов потоковой кластеризации;
- настройки параметров обработки (размер окна, пороги кластеризации и др.);
- визуального контроля процесса сегментации в реальном времени.

Кнопки «Старт» и «Стоп» управляют обработкой: при старте формируется потоковая сессия с загрузкой данных, инициализацией алгоритмов и запуском расширяющегося окна; вычисления вынесены в отдельный поток для предотвращения блокировки интерфейса.

При активизации панели расширенных параметров открывается возможность ручной корректировки обработки и кластеризации: основных настроек потока данных и параметров алгоритмов CluStream и DenStream. В коде эти значения собираются в словари параметров и передаются в алгоритмы только при включённом режиме пользовательских настроек.

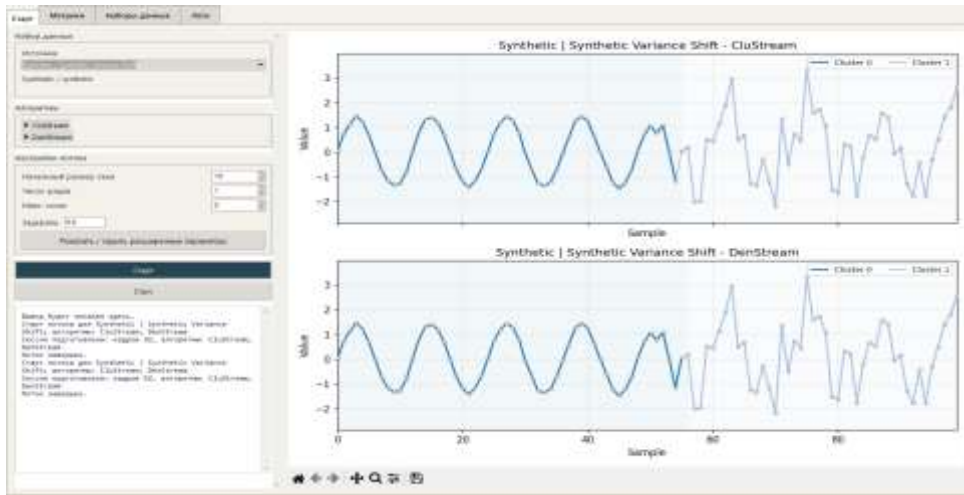


Рис. 1. Основной экран программного приложения
 Fig. 1. The main screen of the software application

Таблица. Усреднённые показатели качества сегментации
 Table. Average segmentation quality indicators

Алгоритм Algorithm	Boundary MAE	Max error worst	FAR	FRR	Accuracy	Recall	Precision	Число кластеров Number of clusters	Purity	Score
CluStream	3,026	38,0	0,121	0,207	0,866	0,793	0,538	2,688	0,933	0,840
DenStream	2,333	29,0	0,155	0,169	0,843	0,831	0,489	3,093	0,963	0,846

Экспериментальные исследования

Для проверки работы алгоритмов использовались синтетические и реальные ВР.

а) Синтетические данные имеют заранее известные границы режимов, что важно для оценки качества сегментации и пределов возможностей алгоритмов. При исследовании использованы восемь синтетических сценариев изменений ВР: среднего уровня; дисперсии; амплитуды; кратковременный всплеск; частоты и сочетание нескольких типов изменений ВР одновременно. В каждом ВР содержится 100 отсчетов. В качестве шума выбран нормально распределённый шум с нулевым средним и заданным стандартным отклонением σ .

Качество сегментации оценивалось с помощью метрик обнаружения границ. При предсказании границы использована зона допуска, поскольку потоковая обработка не гарантирует точного совпадения в момент перехода. Для оценки качества использованы следующие метрики [17]:

Boundary MAE – среднее расстояние между истинными и предсказанными границами (в отсчетах), характеризует точность локализации;

- Max error worst – наибольшая ошибка среди всех тестов, показывает худший случай;
- FAR – доля ложных срабатываний (лишних границ);
- FRR – доля пропущенных границ;

- Accuracy – доля корректно классифицированных точек (граница/не граница);
- Recall – доля обнаруженных истинных границ;
- Precision – доля найденных границ, действительно являющихся истинными.
- Также использованы специфические оценки качества кластеризации в машинном обучении:
- Purity – показывает, насколько точно алгоритм сгруппировал объекты по их истинным классам;
- Score.

Для получения устойчивых показателей качества сегментации было проведено 50 экспериментов по каждому из 8 тестовых сценариев. Усредненные по сценариям и экспериментам метрики для алгоритмов потоковой кластеризации приведены в таблице.

Анализ показателей качества позволил сравнить эффективность алгоритмов кластеризации. По средней ошибке локализации границ алгоритм DenStream превосходит CluStream (Boundary MAE: 2,333/3,026 соответственно), а также имеет меньшую максимальную ошибку (29/38). По полноте DenStream также лучше: Recall выше (0,831/0,793), FRR ниже (0,169/0,207), что указывает на более редкие пропуски переходов. Это объясняется плотной природой алгоритма, чувствительной к появлению новых режимов.

Однако по ложным срабатываниям CluStream эффективнее: FAR ниже (0,121/0,155), Precision выше

(0,538/0,489), и он формирует меньше кластеров (2,688/3,093), давая более консервативную и компактную сегментацию.

Таким образом, алгоритм DenStream точнее обнаруживает реальные переходы, однако алгоритм CluStream реже ошибается с лишними границами. Для окончательных выводов требуется анализ по отдельным сценариям.

б) Реальные временные ряды получены из открытых источников и содержат, в том числе, естественный шум, переходные участки, нестабильные значения и не всегда имеют строго заданные границы между режимами. Поэтому сравнение алгоритмов основывалось не на точности определения момента перехода, а на их способности выделять режимы, различающиеся по характеру сигнала.

Апробация проводилась на реальных данных промышленного мониторинга (расход жидкости и вибрация насоса) и серверной телеметрии (загрузка CPU EC2).

ВР вибрации насоса содержит три характерных участка: нормальный режим (низкий уровень),

аномалия (резкий рост и повышенные значения) и возврат к спокойному состоянию (рис. 2).

Данный пример показателен для задачи сегментации и выделения режимов: дисбаланс ротора меняет не только уровень, но и характер сигнала. Оба алгоритма верно определили смену режима при дисбалансе ротора. Однако внутри участка повышенной вибрации отдельные ступени были ошибочно интерпретированы как самостоятельные режимы.

Еще один ВР – расход жидкости (л/мин) в насосном контуре – демонстрирует два режима. Начальный участок – высокий стабильный уровень соответствует нормальному режиму циркуляции. Затем, после срабатывания клапана, происходит снижение расхода и переход к новому, более низкому уровню (рис. 3).

Оба алгоритма устойчиво выделили режим нормальной циркуляции и последующий режим пониженного расхода. Возвратные колебания в конце ряда не были интерпретированы как отдельный режим – это допустимо, поскольку в реальных данных возврат к исходному состоянию может быть неполным.

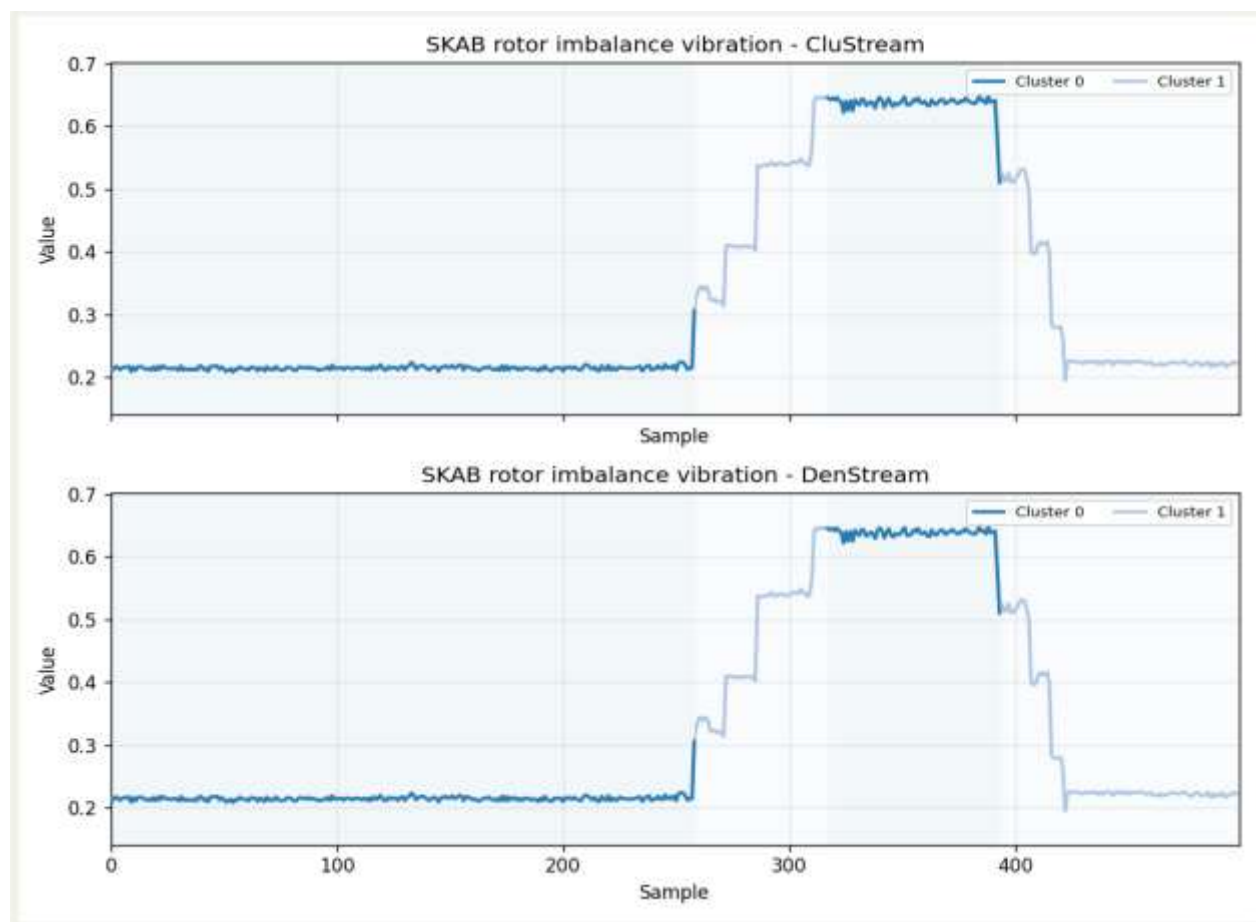


Рис. 2. Результаты работы алгоритмов сегментации при вибрации насоса
Fig. 2. Results of segmentation algorithms for pump vibration

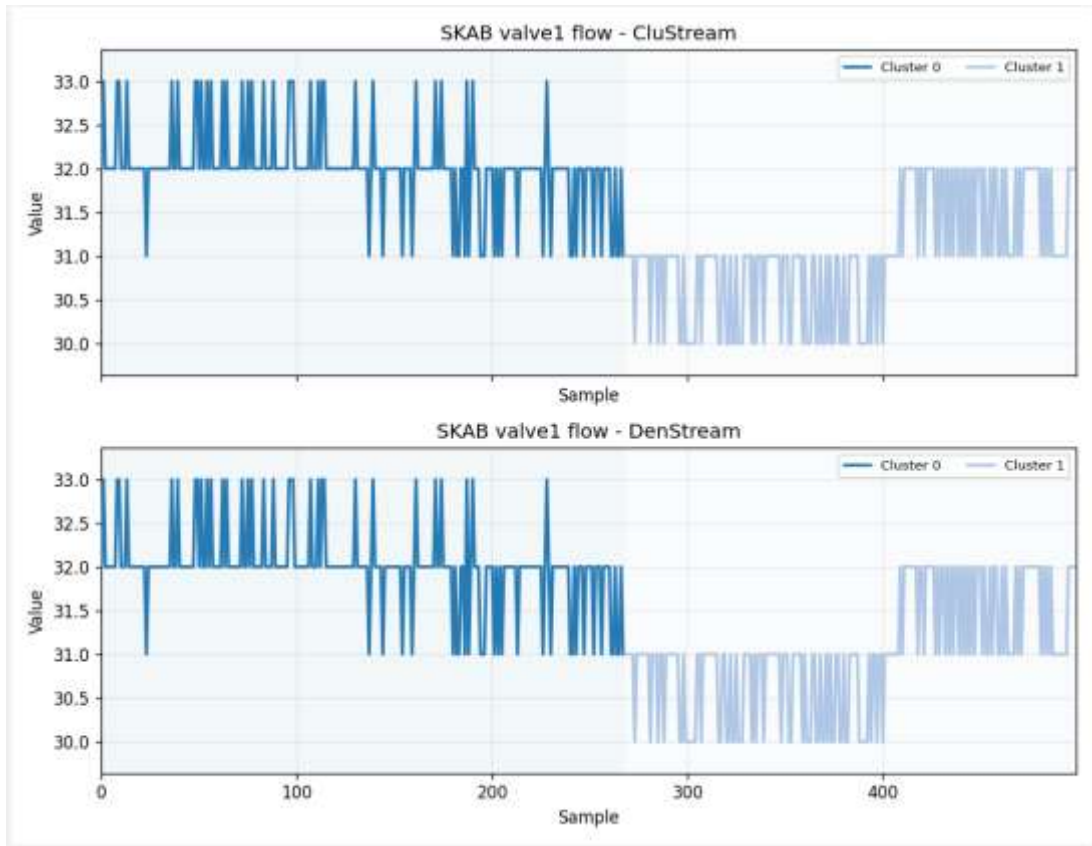


Рис. 3. Результаты работы алгоритмов сегментации в насосном контуре
Fig. 3. Results of segmentation algorithms in a pumping circuit

Заключение

Проведённый анализ подтвердил возможность применения алгоритмов потоковой кластеризации для сегментации ВР в реальном времени. Эксперименты на восьми синтетических сценариях выявили различия: алгоритм DenStream точнее локализует переходы и реже их пропускает, но CluStream даёт меньше ложных границ и формирует более стабильное разбиение.

Результаты на реальных данных оказались менее однозначными, чем на синтетических, из-за шумов, переходных процессов и неполного возврата к

исходному состоянию. Поэтому точное разбиение достигается не всегда. Однако для потоковой обработки важнее, что оба алгоритма устойчиво обнаруживают основные изменения поведения сигнала и представляют ряд как последовательность интерпретируемых режимов.

Дальнейшие исследования будут включать расширение набора признаков, апробацию других алгоритмов потоковой кластеризации, применение к многомерным данным и тестирование на специализированных реальных наборах из различных прикладных областей.

СПИСОК ЛИТЕРАТУРЫ

1. Guijo-Rubio D., Durán-Rosal A.M., Gutiérrez P.A., Troncoso A., Hervás-Martínez C. Time-series clustering based on the characterization of segment typologies. *IEEE Transactions on Cybernetics*, 2020, Vol. 51, № 11, P. 5408–5422. DOI: 10.1109/TCYB.2019.2962584
2. Truong C., Oudre L., Vayatis N. Selective review of offline change point detection methods. *Signal Processing*, 2020, Vol. 167, Article 107299. DOI: 10.1016/j.sigpro.2019.107299 EDN: RCRQZX
3. Fearnhead P., Rigaill G. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 2019, Vol. 114, № 525, P. 169–183. DOI: <https://doi.org/10.1080/01621459.2017.1385466>
4. Xu R., Song Z., Wu J., Wang C., Zhou S. Change-point detection with deep learning: a review. *Frontiers of Engineering Management*, 2025, Vol. 12, № 1, P. 154–176. DOI: 10.1007/s42524-025-4109-z.
5. Tartakovsky A., Nikiforov I., Basseville M. *Sequential analysis: hypothesis testing and changepoint detection*. Boca Raton: CRC Press, 2014. 604 p. DOI: https://doi.org/10.1111/rssa.12122_5
6. Aminikhanghahi S., Cook D.J. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 2017, Vol. 51, № 2, P. 339–367. DOI: <https://doi.org/10.1007/s10115-016-0987-z> EDN: SLNEDS
7. Borovykh A., Bohte S., Oosterlee C.W. Conditional time series forecasting with convolutional neural networks. *International Conference on Artificial Neural Networks*, 2017. DOI: <https://doi.org/10.48550/arXiv.1703.04691>

8. Ren Y., Pu J., Yang Z., Xu J., Li G., Pu X., Yu P.S., He L. Deep clustering: a comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2025, Vol. 36, № 4, P. 5858–5878. DOI: 10.1109/TNNLS.2024.3403155.
9. Chen Y., Li X. Compound sequential change-point detection in parallel data streams. *Statistica Sinica*, 2023, Vol. 33, № 1, P. 453–474. DOI: 10.5705/ss.202020.0508.
10. Aminikhanghahi S., Wang T., Cook D.J. Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, 2019, Vol. 31, № 5, P. 1010–1023. DOI: 10.1109/TKDE.2018.2850347.
11. Aghabozorgi S., Shirkhorshidi A.S., Wah T.Y. Time-series clustering – a decade review. *Information Systems*, 2015, Vol. 53, P. 16–38. DOI: 10.1016/j.is.2015.04.007 EDN: UQEVOJ
12. Silva J.A., Faria E.R., Barros R.C., Hruschka E.R., De Carvalho A.C.P.L.F., Gama J. Data stream clustering: a survey. *ACM Computing Surveys*, 2013, Vol. 46 (1), P. 1–31. DOI: <https://doi.org/10.1145/2522968.2522981>
13. Nordahl C., Boeva V., Grahn H., Persson-Netz M. On evaluation of data stream clustering algorithms: a survey. *IEEE Access*, 2025, Vol. 13, P. 139524–139546. DOI: 10.1109/access.2025.3596435 EDN: GQSIDD
14. Скобцов В.Ю., Новоселова Н.А. Исследование алгоритмов потоковой кластеризации при решении задачи анализа данных телеметрии малых космических аппаратов. *Известия высших учебных заведений. Приборостроение*, 2020, Т. 63, № 11, С. 1003–1011. DOI: 10.17586/0021-3454-2020-63-11-1003-1011 EDN: CNTRCP
15. Kadomskiy K. Using sequential clustering to identification of hybrid timed automata from analog IIoT data. *Problems of Control and Informatics*, 2021, Vol. 66, № 5, P. 31–44. DOI: 10.34229/1028-0979-2021-5-3.
16. Печеный Е.А., Нуриев Н.К., Старыгина С.Д. Самоорганизующаяся кластеризация потока больших данных. *Прикаспийский журнал: управление и высокие технологии*, 2020, № 1 (49), С. 10–20. EDN: HTUKLI
17. Кластеризация временных рядов: приёмы и метрики. *Habr*, 2017. URL: <https://habr.com/ru/articles/334220/> (дата обращения: 10.05.2026).

Информация об авторах

Елена Алексеевна Кочегурова, кандидат технических наук, доцент отделения информационных технологий Инженерной школы информационных технологий и робототехники, Национальный исследовательский Томский политехнический университет, Российская Федерация, 634050, г. Томск, пр. Ленина, 30; kocheg@tpu.ru; <https://orcid.org/0000-0003-4473-528X>

Виталий Владимирович Ленский, студент Инженерной школы информационных технологий и робототехники, Национальный исследовательский Томский политехнический университет, Российская Федерация, 634050, г. Томск, пр. Ленина, 30; vv147@tpu.ru

Поступила: 20.05.2026

Принята: 26.06.2026

Опубликована: 30.06.2026

REFERENCES

1. Guijo-Rubio D., Durán-Rosal A.M., Gutiérrez P.A., Troncoso A., Hervás-Martínez C. Time-series clustering based on the characterization of segment typologies. *IEEE Transactions on Cybernetics*, 2020, vol. 51, no. 11, pp. 5408–5422. DOI: 10.1109/TCYB.2019.2962584
2. Truong C., Oudre L., Vayatis N. Selective review of offline change point detection methods. *Signal Processing*, 2020, vol. 167, article 107299. DOI: 10.1016/j.sigpro.2019.107299 EDN: RCRQZX.
3. Fearnhead P., Rigai G. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 2019, vol. 114, no. 525, pp. 169–183. DOI: <https://doi.org/10.1080/01621459.2017.1385466>
4. Xu R., Song Z., Wu J., Wang C., Zhou S. Change-point detection with deep learning: a review. *Frontiers of Engineering Management*, 2025, vol. 12, no. 1, pp. 154–176. DOI: 10.1007/s42524-025-4109-z.
5. Tartakovsky A., Nikiforov I., Basseville M. *Sequential analysis: hypothesis testing and changepoint detection*. Boca Raton, CRC Press, 2014. 604 p. DOI: https://doi.org/10.1111/rssa.12122_5.
6. Aminikhanghahi S., Cook D.J. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 2017, vol. 51, no. 2, pp. 339–367. DOI: <https://doi.org/10.1007/s10115-016-0987-z> EDN: SLNEDS
7. Borovykh A., Bohte S., Oosterlee C.W. Conditional time series forecasting with convolutional neural networks. *International Conference on Artificial Neural Networks*, 2017. DOI: <https://doi.org/10.48550/arXiv.1703.04691>
8. Ren Y., Pu J., Yang Z., Xu J., Li G., Pu X., Yu P.S., He L. Deep clustering: a comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2025, vol. 36, no. 4, pp. 5858–5878. DOI: 10.1109/TNNLS.2024.3403155.
9. Chen Y., Li X. Compound sequential change-point detection in parallel data streams. *Statistica Sinica*, 2023, vol. 33, no. 1, pp. 453–474. DOI: 10.5705/ss.202020.0508.
10. Aminikhanghahi S., Wang T., Cook D.J. Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, 2019, vol. 31, no. 5, pp. 1010–1023. DOI: 10.1109/TKDE.2018.2850347.
11. Aghabozorgi S., Shirkhorshidi A.S., Wah T.Y. Time-series clustering – a decade review. *Information Systems*, 2015, vol. 53, pp. 16–38. DOI: 10.1016/j.is.2015.04.007 EDN: UQEVOJ
12. Silva J.A., Faria E.R., Barros R.C., Hruschka E.R., De Carvalho A.C.P.L.F., Gama J. Data stream clustering: a survey. *ACM Computing Surveys*, 2013, vol. 46 (1), pp. 1–31. DOI: <https://doi.org/10.1145/2522968.2522981>
13. Nordahl C., Boeva V., Grahn H., Persson-Netz M. On evaluation of data stream clustering algorithms: a survey. *IEEE Access*, 2025, vol. 13, pp. 139524–139546. DOI: 10.1109/access.2025.3596435 EDN: GQSIDD

14. Skobtsov V.Yu., Novoselova N.A. Investigation of stream clustering algorithms when solving the problem of small spacecraft telemetry data analysis. *Journal of Instrument Engineering*, 2020, vol. 63, no. 11, pp. 1003–1011. (In Russ.) DOI: 10.17586/0021-3454-2020-63-11-1003-1011 EDN: CNTRCP
15. Kadomskiy K. Using sequential clustering to identification of hybrid timed automata from analog IIoT data. *Problems of Control and Informatics*, 2021, vol. 66, no. 5, pp. 31–44. DOI: 10.34229/1028-0979-2021-5-3.
16. Pechenyy E.A., Nuriev N.K., Sarygina S.D. Self-organizing clustering of great data flow. *Caspian Journal: Control and High Technologies*, 2020, vol. 1 (49), pp. 10–20. (In Russ.) EDN: HTUKLI
17. Siarshai. Time series clustering: techniques and metrics. *Habr*, 2017. Available at: <https://habr.com/ru/articles/334220/> (accessed: 10 May 2026).

Information about the authors

Elena A. Kochegurova, Cand. Sci. (Eng.), Associate Professor, National Research Tomsk Polytechnic University, 30, Lenin avenue, Tomsk, 634050, Russian Federation; kaicc@tpu.ru; <https://orcid.org/0000-0003-4473-528X>
Vitaliy V. Lenskiy, Student, National Research Tomsk Polytechnic University, 30, Lenin avenue, Tomsk, 634050, Russian Federation; vvl47@tpu.ru

Received: 20.05.2026

Revised: 26.06.2026

Accepted: 30.06.2026