УДК 004.93

***Дымов Александр Сергеевич**, аспирант, Национальный исследовательский университет «МИЭТ», г. Москва*

Dymov Aleksandr Sergeevich, Post-Graduate Student, National Research University of Electronic Technology «MIET», Moscow

e-mail: dymov01as@yandex.ru

СОВРЕМЕННЫЕ МЕТОДЫ УСКОРЕНИЯ АЛГОРИТМА DBSCAN ДЛЯ СИСТЕМ ИНТЕРНЕТА ВЕЩЕЙ

Аннотация: Развитие систем интернета вещей привело к необходимости обработки крупных массивов данных при ограниченных вычислительных ресурсах. Алгоритм плотностной кластеризации DBSCAN эффективно выявляет кластеры произвольной формы и устойчив к шумовым данным. Однако квадратичная сложность алгоритма затрудняет его применение DBSCAN для больших массивов данных. Работа содержит систематический анализ четырех основных направлений ускорения DBSCAN: методы выборочной обработки, пространственного разбиения, случайных проекций и параллельных вычислений. Рассмотрены теоретические основы и практические результаты оптимизационных техник с учетом специфики периферийных вычислений. По данным исследований, количество IoT-устройств в России составило 102,3 миллиона единиц в 2024 году (рост 19% к предыдущему году). Сформулированы рекомендации по выбору методов исходя из характеристик данных и ресурсных ограничений.

Ключевые слова и фразы: DBSCAN, кластеризация, интернет вещей, периферийные вычисления, большие данные, оптимизация алгоритмов, временная сложность

Key words and phrases: DBSCAN, clustering, internet of things, edge computing, big data, algorithm optimization, time complexity

Abstract: The development of Internet of Things systems has led to the need for processing large data arrays with computational resource constraints. The density-based clustering algorithm DBSCAN effectively identifies clusters of arbitrary

shape and is robust to noisy data. However, its quadratic complexity makes DBSCAN difficult to apply to large datasets. This work contains a systematic analysis of four main directions for accelerating DBSCAN: sampling methods, spatial partitioning, random projections, and parallel computing. Theoretical foundations and practical results of applying various optimization techniques are analyzed, taking into account the specifics of edge computing. According to research, the number of IoT devices in Russia reached 102.3 million units in 2024 (19% growth compared to the previous year). Recommendations for method selection based on data characteristics and resource constraints are formulated.

1. Введение

Цифровая трансформация экономики привела к интенсивному росту количества IoT-устройств. Согласно данным Ассоциации интернета вещей и агентства Onside, в 2024 году количество IoT-устройств в России достигло 102,3 миллиона единиц (рост 19% к предыдущему году), объем рынка составил 181 миллиард рублей [1]. Рост числа подключенных устройств создает потребность в эффективных алгоритмах обработки больших объемов данных при ограниченных вычислительных ресурсах периферийных устройств. Кластеризация данных играет ключевую роль в решении этой задачи, позволяя группировать схожие объекты для последующего анализа.

Кластеризация является фундаментальной задачей интеллектуального анализа IoT-данных. Жукова и Аунг Мьо То отмечают, что эффективная кластеризация узлов в динамических IoT-сетях способствует снижению энергопотребления и повышению безопасности сбора данных [2]. Практические применения включают группировку сенсоров по типам данных, выявление аномалий в потреблении энергии и оптимизацию маршрутизации в сенсорных сетях.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) был предложен Эстером и соавторами в 1996 году [3]. Алгоритм способен обнаруживать кластеры произвольной геометрической формы и эффективно обрабатывать шумовые данные. Это критично для IoT-приложений, где

сенсоры могут давать сбои, а структура данных содержит нестандартные паттерны.

Основным препятствием применения DBSCAN остается квадратичная временная сложность $O(n^2)$ в худшем случае. Для датасета из миллиона точек это означает порядка 10^{12} операций сравнения расстояний. При этом периферийные IoT-устройства имеют жесткие ограничения по вычислительным ресурсам и энергопотреблению.

Традиционные методы кластеризации сталкиваются с «проклятием размерности» в высокоразмерных пространствах, характерных для IoT-данных [4]. Выделяют пять ключевых вызовов больших данных: объем (Volume), скорость (Velocity), разнообразие (Variety), достоверность (Veracity) и ценность (Value) [5].

Целью настоящего исследования является систематический анализ современных подходов к ускорению DBSCAN, оценка их применимости для IoT-систем и выявление перспективных направлений развития области.

2. Теоретические основы плотностной кластеризации

2.1 Принципы работы DBSCAN

DBSCAN строится на идее локальной плотности. Алгоритм использует два параметра: радиус окрестности ε и минимальное количество точек $minPts$ для формирования плотной области.

Каждая точка x имеет ε-окрестность $N_\varepsilon(x) = \{y \in D \mid dist(x,y) \le \varepsilon\}$, где D представляет исходный набор данных. Точка классифицируется как ядерная (core point), если $|N_\varepsilon(x)| \ge minPts$. Неядерные точки в ε-окрестности ядерной становятся граничными (border points). Остальные – шум (noise). Кластер формируется путем объединения плотностно-связанных ядерных точек. К ним добавляются граничные точки. Шумовые точки остаются вне кластеров.

Преимущество DBSCAN перед центроидными методами вроде k-means заключается в способности обнаруживать кластеры произвольной формы без предварительного задания их количества. Алгоритм устойчив к выбросам, что важно для IoT-данных с возможными ошибками измерений.

2.2 Вычислительные ограничения

Основной вычислительной операцией DBSCAN является поиск соседей в ε-окрестности. Наивная реализация требует сравнения каждой точки со всеми остальными, что дает $O(n^2)$ операций. На практике для ускорения обычно применяются пространственные индексы (R-дерево, kd-дерево), которые снижают среднюю сложность до $O(n \log n)$.

Однако эффективность индексов существенно снижается в многомерных пространствах. Согласно правилу для k-d деревьев, необходимо $N \gg 2^k$ точек для поддержания эффективности, где k - размерность. При размерности свыше 15-20 поиск деградирует до уровня полного перебора. IoT-данные часто характеризуются высокой размерностью: временные ряды от множественных сенсоров, спектральные характеристики, многопараметрические измерения.

Пространственная сложность составляет $O(n)$ для хранения точек и их статусов. Добавление индексных структур увеличивает требования к памяти, что ограничивает размер обрабатываемых данных на устройствах с лимитированной памятью.

Выбор параметров ε и $minPts$ представляет ещё одну проблему. Неоптимальные значения приводят к фрагментации естественных кластеров или неправомерному объединению. Кроме того, для потоковых IoT-данных с изменяющейся статистикой статические параметры часто неэффективны.

2.3 Развитие плотностных методов

М. Анкерст и соавторы в 1999 году представили OPTICS (Ordering Points To Identify the Clustering Structure. Алгоритм упорядочивает точки по достижимому расстоянию, что позволяет извлекать кластеры для различных значений ε без пересчёта [6]. OPTICS решает проблему выбора параметров, но остается медленным.

HDBSCAN, предложенный Кампелло и коллегами в 2013 году, объединяет плотностную и иерархическую кластеризацию [7]. Алгоритм автоматически определяет стабильные кластеры из иерархии плотностей. HDBSCAN хорошо работает с данными переменной плотности.

Важным преимуществом HDBSCAN является улучшенная вычислительная эффективность по сравнению с классическим DBSCAN за счет оптимизированных алгоритмов построения минимального остовного дерева. Однако алгоритм остается вычислительно затратным для очень больших данных из-за необходимости построения иерархии кластеров.

Несмотря на улучшения в автоматическом выборе параметров, оба подхода остаются вычислительно затратными. Для IoT-приложений требуются методы, ускоряющие базовый DBSCAN без существенного усложнения алгоритма.

3. Современные подходы к ускорению DBSCAN

Растущие требования к обработке больших данных стимулировали разработку множества оптимизационных техник для DBSCAN. Классификация методов по принципу работы позволяет выделить четыре основных направления: выборочная обработка, пространственные индексы, размерностная редукция и параллелизм.

3.1 Методы выборочной обработки

Значительным достижением в области ускорения плотностной кластеризации стал алгоритм DBSCAN++, использующий стратегию выборочной обработки ядерных точек. Вместо анализа всех n точек алгоритм сначала выбирает подмножество S из m точек с помощью равномерной выборки или жадной инициализации k-центров [8].

Основная идея основана на избыточности точек в плотных областях для определения структуры кластеров. Достаточно найти репрезентативные ядерные точки среди выборки, а затем отнести остальные точки к ближайшим кластерам.

Временная сложность DBSCAN++ составляет $O(mn)$, где $m \ll n$. Построение множества S требует $O(n)$ времени для равномерной выборки или $O(mn)$ для жадного подхода. Поиск ядерных точек выполняется через KDTree запросы с $O(n)$ операций на каждую из m точек выборки. Финальное присоединение оставшихся точек к ближайшим ядрам также требует $O(mn)$ операций.

При выборе размера выборки m существенно меньшего n достигается практическое ускорение. Авторы экспериментально показали эффективность различных стратегий выборки, но не предоставили теоретических рекомендаций по оптимальному размеру m. Выбор зависит от распределения данных и требований к качеству кластеризации. Авторы отмечают снижение чувствительности к выбору параметров ε и $minPts$ благодаря устойчивости выборочных методов к локальным вариациям плотности.

Основное ограничение проявляется на данных с сильно неравномерной плотностью. Случайная выборка может пропустить важные разреженные области, что критично для IoT-сенсоров с нерегулярным пространственным распределением.

3.2 Пространственно-ориентированные методы

Традиционные индексы R-дерево и kd-дерево эффективны только до 10-15 измерений. Хуанг и соавторы в 2023 году разработали GriT-DBSCAN с древовидной сеточной структурой [9]. Ключевая инновация – адаптивное разбиение пространства на основе локальной плотности данных.

Алгоритм строит иерархическую сетку, где плотные области разбиваются мельче. Это позволяет достичь почти линейной сложности $O(n \log n)$ для реальных датасетов. Эксперименты на синтетических и реальных пространственных данных показали результаты, сопоставимые с классическим DBSCAN при существенном ускорении.

Однако метод требует построения индекса перед кластеризацией, что неприемлемо для потоковых данных. Кроме того, эффективность grid-based подходов снижается при высокой размерности данных. Для многомерных IoT-данных (спектры, временные ряды) требуются другие подходы.

3.3 Методы случайных проекций

Важным развитием в области масштабируемой плотностной кластеризации стал алгоритм sDBSCAN, предложенный Сюй и Фамом в 2024 году на конференции NeurIPS [10]. Метод основан на использовании

случайных проекций для быстрой идентификации кандидатов в ε-окрестности с помощью экстремальной порядковой статистики.

Технической инновацией sDBSCAN является использование большого количества случайных векторов проекции в сочетании с быстрым преобразованием Адамара для эффективного вычисления скалярных произведений. Теоретически, с высокой вероятностью алгоритм сохраняет структуру кластеризации DBSCAN при мягких условиях.

Экспериментальные результаты показывают значительное превосходство над конкурентами. На реальных датасетах с миллионами точек sDBSCAN вместе с sOPTICS (масштабируемая версия OPTICS для визуализации структуры кластеров и подбора параметров sDBSCAN) выполняются за несколько минут, в то время как аналоги из scikit-learn требуют нескольких часов или не могут выполниться из-за ограничений памяти. Алгоритм также предоставляет более высокую точность по сравнению с другими вариантами DBSCAN.

Особенностью sDBSCAN является специализация на высокоразмерных данных с косинусной метрикой расстояния. Авторы расширили алгоритм на евклидову, манхэттенскую, хи-квадрат и расстояние Йенсена-Шеннона через случайные ядерные признаки. Для IoT-приложений с нормализованными признаками такой подход может обеспечить существенные преимущества в производительности.

3.4 Параллельные реализации

Ван, Гу и Шун в 2019 году предложили эффективный подход к параллелизации DBSCAN [11]. Предложенный алгоритм демонстрирует теоретически оптимальную рабочую сложность, соответствующую лучшим последовательным алгоритмам, при полилогарифмической глубине параллелизма.

Практические результаты показывают возможность достижения 33-кратного ускорения на 36-ядерных системах. Для IoT-приложений

параллельные реализации особенно актуальны в контексте туманных вычислений, где доступны многоядерные серверы промежуточного уровня.

Ограничением является необходимость в shared memory архитектуре. Распределенные версии для Apache Spark показывают худшую эффективность из-за накладных расходов на коммуникацию между узлами.

3.5 Гибридные методы

Комбинирование различных техник ускорения представляет перспективное направление. Цю и соавторы в 2025 году применили гибридный подход Mini-Batch K-means + DBSCAN в составе алгоритма Brain Storm Optimization для задачи планирования пути мультироботных систем [12]. Предварительная кластеризация Mini-Batch K-means группирует точки задач, затем DBSCAN уточняет границы кластеров и выявляет шумовые данные. Гибридный алгоритм HC-BSO показал сокращение времени вычислений на 98,98% (с 75,79 до 0,77 секунд) по сравнению с использованием обычного K-means в алгоритме BSO.

Ин и коллеги в 2024 году в обзоре алгоритмов кластеризации отмечают растущий интерес к гибридным методам [13]. Комбинирование позволяет использовать преимущества разных подходов: быстроту центроидных методов и точность плотностных алгоритмов.

4. Гибридные подходы к кластеризации

4.1 Теоретические основы комбинированных методов

Центроидные и плотностные методы имеют разные характеристики качества производительности и качества. Например, K-means выполняет глобальное разбиение за $O(nkti)$, где n – количество точек, k – количество кластеров, t – количество итераций и i – размерность данных. DBSCAN, в свою очередь, обеспечивает локально точные результаты за $O(n^2)$ в худшем случае без использования индексных структур. Комбинирование этих методов позволяет использовать быстроту K-means для предварительного разбиения и точность DBSCAN для финальной кластеризации в полученных группах.

Математическое обоснование данного гибридного подхода базируется на принципе «разделяй и властвуй». Если K-means создаёт k групп размером n/k каждый, то применение DBSCAN к каждой из них требует $O(k*(n/k)^2) = O(n^2/k)$ операций. При $k = \sqrt{n}$ достигается сложность $O(n^{1.5})$ для этапа с DBSCAN, что существенно лучше квадратичной сложности стандартного DBSCAN.

Важным условием эффективности является качество начального разбиения. Если K-means ошибочно объединит различные плотностные кластеры в один, DBSCAN не сможет их разделить на втором этапе. Наоборот, излишнее дробление увеличит вычислительные затраты без улучшения результатов.

4.2 Практические реализации

Сридеви и Раджанна в 2025 году исследовали комбинацию Mini-Batch K-means с DBSCAN на датасете AOL User Session Collection, содержащем 20 миллионов веб-запросов от 650 тысяч пользователей [14]. Первый этап группирует синтаксически схожие запросы в 5 кластеров с помощью Mini-Batch K-means. Второй этап применяет DBSCAN к каждому кластеру для выявления семантических подгрупп и фильтрации шума.

Результаты показали преимущества гибридного подхода: силуэтный коэффициент составил 0,7214 против 0,65 у K-means. Скорректированный индекс Рэнда достиг 0,7823, что указывает на высокое качество разбиения. Авторы отмечают, что гибридный метод обеспечивает компромисс между скоростью и качеством кластеризации для больших текстовых датасетов.

Митин и Панов предложили модификацию DBSCAN для обработки потоковых данных с использованием адаптивных фреймов [15]. Алгоритм применяет гибридный подход к поиску границ кластеров, что снижает требования к памяти по сравнению с классическим DBSCAN. Метод ориентирован на работу с динамическими данными, где накопление полной истории невозможно из-за ресурсных ограничений, поэтому алгоритм периодически перестраивает K-means модель и применяет инкрементальную версию DBSCAN к новым данным.

4.3 Выбор параметров

Выбор количества кластеров k для предварительного разбиения остается открытой задачей. Малые значения k не обеспечивают существенного ускорения, крупные увеличивают накладные расходы. Эмпирические правила вроде $k \approx \sqrt{n}$ часто дают хорошие результаты на практике для некоторых типах данных, но теоретическое обоснование оптимального k требует дальнейших исследований.

Для потоковых систем, где статистические характеристики меняются со временем, особенно важна адаптивность к изменению распределения и характеристик потока. Однако автоматизация настройки гибридных методов остается сложной задачей из-за необходимости одновременной оптимизации параметров обоих алгоритмов.

5. Специфика систем интернета вещей

5.1 Архитектуры периферийных вычислений

IoT-экосистемы требуют иерархической обработки данных. Као и Вачович описывают трехуровневую модель: edge-fog-cloud [16]. На уровне edge выполняется первичная фильтрация и агрегация, в fog – кластеризация и анализ паттернов, в cloud – глобальная аналитика.

Методы кластеризации применяются для эффективного управления данными в мобильных IoT-сетях. Энергоэффективная маршрутизация достигается за счет оптимального выбора центральных узлов кластеров и минимизации передачи данных на дальние расстояния. Такой подход снижает энергопотребление по сравнению со случайным группированием и обеспечивает расширение срока службы сети.

5.2 Ресурсные ограничения edge-устройств

Типичные IoT-контроллеры имеют ограниченные вычислительные ресурсы, включая память (обычно менее 1 ГБ), процессор и энергопитание. Для таких устройств критичны время отклика и энергопотребление.

Оптимизация памяти достигается использованием инкрементальных алгоритмов кластеризации, которые обрабатывают данные по мере

поступления без необходимости хранения полной истории. Такие алгоритмы поддерживают только центроиды кластеров и агрегированные статистики вместо хранения всех точек данных, что существенно снижает требования к памяти. Снижение вычислительной сложности обеспечивается аппроксимационными методами, адаптивными стратегиями выборки и использованием эффективных структур данных.

Энергоэффективность требует минимизации интенсивных вычислений. Использование целочисленной арифметики или фиксированной точности (fixed-point arithmetic) вместо операций с плавающей запятой позволяет продлить автономность устройств.

5.3 Перспективы развития

Российский рынок IoT демонстрирует устойчивый рост как по количеству подключенных устройств, так и в денежном выражении [1]. Это создает спрос на эффективные алгоритмы обработки данных, адаптированные к специфике IoT-систем.

Глобальный рынок периферийных вычислений демонстрирует динамичное развитие. По данным IDC, мировые расходы на edge computing достигнут 261 миллиарда долларов в 2025 году с прогнозируемым среднегодовым ростом 13,8%, что создает благоприятные условия для внедрения алгоритмов кластеризации в IoT-системах [17].

Развитие высокоскоростных сетей 5G и NB-IoT создает новые возможности для периферийных вычислений с низкими задержками. Интеграция с методами машинного обучения открывает перспективы для адаптивной кластеризации с автоматическим подбором параметров.

6. Сравнительный анализ методов

6.1 Критерии оценки эффективности

Для объективного сравнения методов ускорения DBSCAN необходимо учитывать временную сложность, качество кластеризации, масштабируемость и применимость к различным типам данных. Потребление памяти критично для IoT-устройств с ограниченными ресурсами.

Качество кластеризации оценивается внутренними метриками. Силуэтный коэффициент, предложенный Руссеу в 1987 году [18], измеряет компактность внутри кластеров и разделенность между ними. Согласно общепринятой интерпретации, значения выше 0,7 считаются сильными, выше 0,5 – разумными, выше 0,25 – слабыми. Индекс Дэвиса-Боулдина (1979) [19] оценивает отношение внутрикластерных и межкластерных расстояний – меньшие значения указывают на лучшее разделение.

Арбелаитц и коллеги в 2013 году провели обширное сравнительное исследование 30 индексов валидности кластеров [20]. Силуэтный индекс показал наилучшие общие результаты среди всех протестированных индексов, превысив 50% успешных оценок числа кластеров, хотя авторы отмечают отсутствие универсального решения для всех типов данных. Для IoT-задач важна высокая интерпретируемость кластеризации, и силуэтный индекс удобен в использовании благодаря простой визуализации качества кластеров.

6.2 Характеристика подходов

Методы выборочной обработки (DBSCAN++) обеспечивают стабильное ускорение при минимальной потере качества кластеризации. Теоретические гарантии сложности $O(nm)$ делают их привлекательными для больших данных. Недостатком является зависимость от равномерности распределения плотности.

Пространственные индексы демонстрируют высокую производительность на низкоразмерных данных. GriT-DBSCAN показывает существенное ускорение при сохранении точности для пространственных данных в Евклидовом пространстве. Алгоритм достигает почти линейной временной сложности по размеру датасета за счет использования иерархической grid tree структуры. Однако, как и другие grid-based методы, он ограничен в применении к высокоразмерным пространствам из-за проклятия размерности.

Метод случайных проекций sDBSCAN эффективен для высокоразмерных данных. Алгоритм демонстрирует значительное ускорение для больших наборов данных, но специализация на косинусной метрике ограничивает

применимость для IoT-сенсоров, где часто предпочтительна евклидова метрика.

Параллельные реализации показывают хорошую масштабируемость на многоядерных системах. Для fog computing с мощными серверами промежуточного уровня это актуально. Edge-устройства, хотя и обладают несколькими ядрами (обычно 2-8), имеют ограничения по энергопотреблению и вычислительной мощности, что требует специализированных облегчённых реализаций.

Гибридные подходы позволяют комбинировать преимущества различных методов. Например, возможно сочетание K-means и DBSCAN. Недостатком такого подхода является сложность настройки параметров обоих алгоритмов одновременно.

6.3 Рекомендации по выбору метода

Для пространственных IoT-данных от GPS-трекеров или сенсоров окружающей среды рекомендуются методы пространственного индексирования. Они обеспечивают эффективный анализ низкоразмерных данных с умеренным потреблением памяти. Высокоразмерные данные, такие как спектральные признаки или многопараметрические временные ряды, эффективнее обрабатывать с использованием случайных проекций. Для динамических потоковых данных целесообразны гибридные методы с адаптивной настройкой параметров.

В условиях жестких ограничений ресурсов предпочтительны методы выборочной обработки с возможностью динамической настройки размера выборки в зависимости от доступных вычислительных мощностей.

7. Перспективы и направления развития

7.1 Нерешенные проблемы

Автоматический выбор параметров остается ключевой проблемой плотностных методов кластеризации. Большинство исследований предполагают оптимальные значения ε и $minPts$ либо требуют ручной настройки. Для IoT-систем с тысячами устройств это неприемлемо.

Бахтияри и коллеги предложили ClustRecNet – нейросетевой подход к выбору алгоритма кластеризации [24]. Модель обучена на 34,000 синтетических датасетах и показывает улучшение на 15,3% по метрике ARI над лучшими AutoML подходами. Однако модель тестировалась на статических данных и требует адаптации к динамическим IoT-потокам.

Масштабируемость алгоритмов по размерности данных остается недостаточно изученной областью. Хотя многие методы тестируются на размерностях порядка 100-200, современные IoT-системы генерируют высокоразмерные временные ряды, поведение алгоритмов в таких условиях требует дальнейших исследований.

Энергопотребление редко учитывается в академических исследованиях алгоритмов кластеризации. Для автономных IoT-устройств это критично, поскольку вычислительные операции напрямую влияют на время работы от батареи.

7.2 Перспективные направления

Федеративное обучение открывает новые возможности для распределенной кластеризации в IoT-системах. Подход FedDB демонстрирует эффективность DBSCAN в федеративных сетях для решения проблемы неоднородного распределения данных [21]. Каждое устройство выполняет локальную кластеризацию, затем участвует в агрегации результатов без передачи исходных данных, что решает проблемы приватности и снижает сетевую нагрузку.

Исследования в области квантовых алгоритмов кластеризации активно развиваются. Существуют работы по квантовому DBSCAN [22], но практическое применение ограничено текущим уровнем развития квантовых компьютеров. Реальные преимущества пока демонстрируются только на специальных задачах.

Адаптация к изменениям в распределении данных (concept drift) критична для потоковых IoT-данных [23]. Автоматическое обнаружение изменений в

статистике данных и динамическая адаптация параметров кластеризации представляют важное направление исследований.

Специализация алгоритмов под конкретные IoT-домены позволит использовать доменные знания. Кластеризация медицинских, промышленных и транспортных данных имеет разные требования к точности, скорости и энергопотреблению. Разработка domain-specific алгоритмов может существенно улучшить результаты.

8. Заключение

Проведенный систематический анализ современных методов ускорения алгоритма DBSCAN демонстрирует значительный прогресс в решении проблемы масштабируемости плотностной кластеризации. Предложенные подходы, от методов выборочной обработки до гибридных алгоритмов, снижают вычислительную сложность и расширяют возможности применения DBSCAN в IoT-системах.

Анализ показывает отсутствие универсального решения, оптимального для всех типов IoT-данных и применений. Методы выборочной обработки эффективны для больших однородных данных, пространственные индексы – для низкоразмерных пространственных данных, случайные проекции – для высокоразмерных признаковых пространств, параллельные реализации – для многоядерных систем fog computing, а гибридные подходы – для потоковых данных с изменяющимися характеристиками. Выбор оптимального метода должен основываться на анализе характеристик данных, ограничений вычислительных ресурсов и требований к качеству кластеризации.

Рост российского рынка IoT создает потребность в эффективных алгоритмах кластеризации, адаптированных к специфическим требованиям IoT-систем. Ключевые направления дальнейших исследований включают автоматизацию выбора параметров кластеризации, улучшение масштабируемости алгоритмов по размерности данных и адаптацию к динамически изменяющимся характеристикам потоковых данных. Перспективные подходы включают федеративное обучение для

распределенной обработки данных и адаптивные методы для работы с изменениями в распределении данных.

Успешное развитие области требует интеграции теоретических исследований с практическими потребностями IoT-индустрии. Междисциплинарный подход, объединяющий алгоритмику, системный анализ и прикладные аспекты интернета вещей, необходим для создания эффективных решений современных вызовов обработки больших данных.

Литература

1. Исследование российского рынка интернета вещей 2024 / Ассоциация интернета вещей, агентство Onside. URL: https://www.tadviser.ru/index.php/Статья:Интернет_вещей,_IoT_(рынок_России) (дата обращения: 15.10.2025).
2. Zhukova N.A., Aung M.T., Tin T.A., Evnevich E.L. Model based on the selection of cluster heads for data collection in the mobile Internet of Things // Radio Industry (Radiotekhnika). 2020. Vol. 7, No. 4. P. 31–38.
3. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996. P. 226–231.
4. Wani A.A. Comprehensive analysis of clustering algorithms: exploring limitations and innovative solutions // PeerJ Computer Science. 2024. Vol. 10. Article e2286.
5. Awad F.H., Hamad M.M. Big Data Clustering Techniques Challenges and Perspectives: Review // Informatica. 2023. Vol. 47, No. 6. P. 203-218.
6. Ankerst M., Breunig M.M., Kriegel H.-P., Sander J. OPTICS: ordering points to identify the clustering structure // ACM SIGMOD Record. 1999. Vol. 28, No. 2. P. 49–60.
7. Campello R.J.G.B., Moulavi D., Sander J. Density-based clustering based on hierarchical density estimates // Advances in Knowledge Discovery and Data Mining. Berlin: Springer, 2013. P. 160–172.

8. Jang J., Jiang H. DBSCAN++: Towards fast and scalable density clustering // Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. P. 3019–3029.
9. Huang X., Ma T., Liu C., Liu S. GriT-DBSCAN: A spatial clustering algorithm for very large databases // Pattern Recognition. 2023. Vol. 142. Article 109658.
10. Xu H., Pham N. Scalable DBSCAN with Random Projections // Proceedings of the 38th Conference on Neural Information Processing Systems. Vancouver: NeurIPS Foundation, 2024. P. 15642–15658.
11. Wang Y., Gu Y., Shun J. Theoretically-Efficient and Practical Parallel DBSCAN // Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland: ACM, 2020. P. 1493–1508.
12. Qiu G., Deng J., Li J., Wang W. Hybrid Clustering-Enhanced Brain Storm Optimization Algorithm for Efficient Multi-Robot Path Planning // Biomimetics. 2025. Vol. 10, No. 6. Article 347. DOI: 10.3390/biomimetics10060347.
13. Yin H., Aryani A., Petrie S., Nambissan A., Astudillo A. A Rapid Review of Clustering Algorithms // arXiv preprint arXiv:2401.07389. 2024.
14. Sridevi K.N., Rajanna M. Hybrid Clustering Framework for Scalable and Robust Query Analysis: Integrating Mini-Batch K-Means with DBSCAN // International Journal of Advanced Computer Science and Applications. 2025. Vol. 16, No. 1. P. 673–683.
15. Митин Г.В., Панов А.В. Модификация алгоритма DBSCAN с использованием гибридных подходов к определению границ кластеров для обработки потоковых данных // IT-Стандарт (IT-Standard). 2024. Т. 21, № 3. С. 45–52. URL: https://itstd-journal.ru/wp-content/uploads/2024/03/MODIFICATION-OF-DBSCAN-ALGORITHM-USING-.pdf

16. Cao H., Wachowicz M. An Edge-Fog-Cloud Architecture of Streaming Analytics for Internet of Things Applications // Sensors. 2019. Vol. 19, No. 16. Article 3594.
17. IDC Estimates Global Spending on Edge Computing to Grow at 13.8% Reaching Nearly $380 Billion by 2028. Needham: International Data Corporation, 2025. URL: https://www.idc.com/getdoc.jsp?containerId=prUS53261225 (дата обращения: 30.10.2025).
18. Rousseeuw P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis // Journal of Computational and Applied Mathematics. 1987. Vol. 20. P. 53–65.
19. Davies D.L., Bouldin D.W. A Cluster Separation Measure // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1979. Vol. PAMI-1, No. 2. P. 224–227.
20. Arbelaitz O., Gurrutxaga I., Muguerza J., Pérez J.M., Perona I. An extensive comparative study of cluster validity indices // Pattern Recognition. 2013. Vol. 46, No. 1. P. 243–256.
21. Lee Y.-C., Chien W.-C., Chang Y.-C. FedDB: A Federated Learning Approach Using DBSCAN for DDoS Attack Detection // Applied Sciences. 2024. Vol. 14, No. 22. Article 10236.
22. Xie X., Duan L., Qiu T., Li J. Quantum algorithm for MMNG-based DBSCAN // Scientific Reports. 2021. Vol. 11. Article 15559.
23. Chu R., Jin P., Qiao H., Feng Q. Intrusion detection in the IoT data streams using concept drift localization // AIMS Mathematics. 2024. Vol. 9, No. 1. P. 1535–1561.
24. Bakhtyari M., Mazoure B., Cordeiro de Amorim R., Rabusseau G., Makarenkov V. ClustRecNet: A Novel End-to-End Deep Learning Framework for Clustering Algorithm Recommendation // arXiv preprint arXiv:2509.25289. 2025.

References

1. Issledovanie rossijskogo rynka interneta veschej 2024 / Associacija interneta veschej, agentstvo Onside. Available at: https://www.tadviser.ru/index.php/Статья:Интернет_вещей,_IoT_(рынок_России) (accessed: 15.10.2025).
15. Mitin G.V., Panov A.V. Modifikacija algoritma DBSCAN s ispol'zovaniem gibridnyh podhodov k opredeleniyu granic klasterov dlya obrabotki potokovih dannyh // IT-Standard. 2024. Vol. 21, No. 3. P. 45–52. Available at: https://itstd-journal.ru/wp-content/uploads/2024/03/MODIFICATION-OF-DBSCAN-ALGORITHM-USING-.pdf