DOI 10.37882/2223-2966.2025.06.46

САМООБУЧАЕМОЕ ФОРМИРОВАНИЕ МЕТРИК КАЧЕСТВА ДАННЫХ ДЛЯ НЕИЗУЧЕННЫХ ПОТОКОВ

SELF-LEARNING FORMATION OF DATA QUALITY METRICS FOR UNEXPLORED STREAMS

K. Ulanov

Summary. The article is devoted to the problem of automatic quality control of «dark» data streams — Kafka topics, for which there is no reference markup and a pre-known scheme. The aim of the work is to develop a self—learning method for generating metrics for the quality of streaming data, capable of evaluating the reliability of unexplored events in real time without manual rules. A streaming algorithm is proposed in which a lightweight online encoder extracts features, a Boolean augmentation mask creates positive and negative examples, and a loss rank function is trained on the principle of self-learning ranking. On the NYC Taxi open set, the method was ahead of the rule-based tests, Isolation Forest and Deep SVDD: the P1000 increased to 0.74, and the error detection delay decreased to 32 seconds when loading 0.55 vCPU. The findings confirm that self-learning ranking is an effective and resource-saving framework for end-to-end data quality control in streaming systems.

Keywords: streaming data processing, data quality control, self-supervised learning, rank learning, Apache Kafka, unexplored streams.

Уланов Кирилл Анатольевич
аспирант, Московский государственный технологический университет «Станкин»
ulanovk08@gmail.com

Аннотация. Статья посвящена проблеме автоматического контроля качества «тёмных» потоков данных — Kafka-топиков, для которых отсутствует эталонная разметка и заранее известная схема. Цель работы — разработать метод самообучаемого формирования метрики качества потоковых данных, способный в реальном времени оценивать достоверность неизученных событий без ручных правил. Предлагается потоковый алгоритм, в котором лёгкий онлайн-энкодер извлекает признаки, булева маска аугментаций создаёт позитивные и негативные примеры, а ранговая функция потерь обучается по принципу самообучающегося ранжирования. На открытом наборе NYC Taxi метод опередил rule-based тесты, Isolation Forest и Deep SVDD: P1000 выросла до 0,74, а задержка обнаружения ошибок сократилась до 32 с при загрузке 0,55 vCPU. Выводы подтверждают, что самообучающееся ранжирование является эффективной и ресурсосберегающей основой для сквозного контроля качества данных в потоковых системах.

Ключевые слова: потоковая обработка данных, контроль качества данных, self-supervised learning, ранговое обучение, Apache Kafka, неизученные потоки.

Введение

В условиях современного роста объёма оперативных данных, проходящих через стриминговые шины Apache Kafka, и обработчики Apache Flink, растёт экспоненциально. По оценкам аналитиков, в 2024 году количество активных Kafka-топиков, используемых бизнесом, превысило 10 миллионов, а ключевым трендом назвали data contracts, что сильно снижает порог подключения новых потоков данных без предварительного аудита качества [1].

Но переход к архитектурам «stream-first» обнаруживает важную проблему: больше половины поступающих сообщений остаются «тёмными данными» — то есть не попадают в каталоги или тесты валидации. Исследования говорят, что 60–85 % неструктурированных файлов и событий хранятся без какой-либо семантической метки или правил жизненного цикла [2]. При этом каждая ошибка в неизученном потоке данных способна мгновенно тиражироваться в аналитические витрины и модели машинного обучения, провоцируя каскад различных инцидентов.

Возрастание масштабов и стоимости ошибок заставила команды разработки и внедрения сместить контроль качества данных на более ранние стадии и одновременно обеспечить наблюдение за качеством данных на пререлизной и пострелизной стадии. Согласно обзору CNCF 2025, главным драйвером стала интеграция алгоритмов машинного обучения, способных автоматически выявлять и ранжировать аномалии в миллиардах потоков данных, логов и метрик [3]. Тем не менее существующие инструменты контроля качества данных (Great Expectations, Deequ, SodaSQL и др.) по-прежнему требуют явных правил и контрольных данных, что делает их неприменимыми к «неизученным» потокам без схем и разметки.

Гипотеза и цель исследования

Мы предполагаем, что самообучающееся ранжирование может заменить ручное создание метрик качества данных при работе с неизученными потоками. Ключевые наблюдения:

В задачах предварительной очистки данных метод SelfClean (NeurIPS 2024) показал, что комбинация

самоконтролируемых признаков и ранжирования по дистанциям позволяет без использования разметки детектировать до 16 % шумовых данных, приближая эффективность к ручной разметке ошибочных данных [4].

Исследования демонстрируют, что смещение оценки качества данных из задачи регрессии в задачу парного ранжирования улучшает обобщающую способность моделей [5].

Это позволяет выдвинуть гипотезу: если из скользящего окна потока данных формировать псевдо-позитивные и негативные пары, то модель ранжирования сможет автоматически выявлять релевантные признаки и вычислять интегральный показатель качества данных в режиме реального времени, не используя заранее заданных правил.

Целью статьи является разработка и верификация метода самообучаемого формирования метрик качества данных для неизученных потоков данных, основанного на самообучающемся ранжировании и интегрируемого в стек Apache Kafka.

Анализ современного состояния проблемы

Для подробного исследования нам необходимо рассмотреть существующие методы по оценке качества потоков данных.

Обзор стандартных фреймворков контроля качества данных

Great Expectations. Фреймворк изначально строился вокруг пакетной обработки данных: пользователю необходимо подготовить «партию» данных, задать ожидания и выполнить валидацию. Поддержка потоков реализуется только в виде «микропакетов» либо собственных адаптеров Kafka/Flink, причём необходимо вручную сформулировать проверки для каждой новой схемы или окна — что принципиально не решает задачу «неизученных» потоков, где заранее нет информации о схеме [6].

Amazon Deequ. Библиотека, встроенная в Spark, автоматически выводит кандидатов-метрик и проверяет их на следующих партиях данных. Несмотря на то, что Spark Structured Streaming допускает инкрементальный расчёт, Deequ фиксирует состояние метрик и потребляет поток серии микропакетов. Для каждой колонки требуется явное правило, то есть при появлении неизвестного столбца система выдаст ошибку [7–8].

Evidently AI. Изначально инструмент создавался для мониторинга моделей машинного обучения. Во «встроенном» модуле Data Drift сравнивается текущий поток признаков с эталонным распределением и сигнализирует о статистических сдвигах. Однако при отсутствии эталонных данных Evidently не может подтвердить сам факт ошибок, а лишь покажет «подозрительное» отличие [9–10].

Современные фреймворки качества данных предоставляют богатый инструментарий для детерминированных схем, но общим ограничением остаётся предпосылка «известной» структуры и/или наличия эталонных данных. В контексте потоков реальных данных число таких «неизученных» топиков ежегодно растёт, а быстро писать проверки вручную невозможно. Следовательно, нужен подход, где метрика формируется напрямую из потока без внешней разметки и постоянного обновления правил.

Самообучающиеся подходы к потоковым данным

Парадигма самообучающихся представлений кажется очень подходящей. Модель извлекает информативные признаки из неразмеченных данных, постепенно генерируя псевдо-задачу. В компьютерном зрении лидируют SimCLR, MoCo, BYOL; в последние два года эти идеи активно переносятся на временные ряды.

SimCLR (Simple Framework for Contrastive Learning of Visual Representations). Метод обучения представлений, разработанный исследователями из Google Research. Метод позволяет обучать нейросетевую модель, извлекать полезные признаковые представления изображений без применения размеченных данных, что особенно важно в условиях ограниченного количества размеченных примеров. SimCLR основывается на контрастивном обучении. Целью является сближение векторных представлений (эмбеддинги) похожих объектов (положительных пар) и отдаление представлений непохожих объектов (отрицательных пар).

Для применения метода к временным рядам требуются доменно-специфичные аугментации. Исследование TS-TCC показало, что модели на основе SimCLR уступают CPC/TS-TCC при учёте временной структуры [11].

BYOL (Bootstrap Your Own Latent) — это метод самообучения без использования контрастивной функции потерь и без необходимости в отрицательных примерах, в отличие от SimCLR. Основная идея BYOL: модель может научиться полезным признаковым представлениям, просто сравнивая свои собственные прогнозы с собой — без явного сопоставления с другими образцами. Особенностями применения к временным рядам является то, что метод экономит память. Метод устойчив к ситуации, когда модель теряет разнообразие векторных представлений и «схлопывает» всё в одно (или очень похожие) представления, но на ранних этапах требует длинного окна стабилизации, что усложняет вычисления в реальном времени [12].

RankSim. Метод, относящийся к классу ранговых подходов. Его ключевая идея — использовать структуру относительных расстояний между представлениями, а не сами представления напрямую. Изначально метод был предложен для несбалансированных регрессионных задач. Современные версии (W-RankSim, Rank-Me) показывают, что ранговые регуляризаторы устойчивы к микропакетной обработке и могут дать интегральную оценку качества данных без явной разметки [13–14].

Заметно развивается линия работ TS2Vec, TF-C и Dynamic Contrastive Learning, где контекстно-зависимые позитивы формируются по спектру частот или множеству временных масштабов, обеспечивая лучшее обобщение на нестатичные ряды [15].

Контрастивные методы (SimCLR-семейство) демонстрируют высокую точность при богатом наборе доменных аугментаций, но чувствительны к размеру батча и числу негативных примеров. Предиктивные модели CPC/TS-TCC предсказывают будущее состояние ряда и потому устойчивы к сдвигу фаз, однако требуют рекуррентных/трансформерных энкодеров с высокой задержкой. Ранговые регуляризаторы (RankSim, Rank-Me) переводят оценку в порядковое пространство, что логично для задач качества данных. Тем не менее ни один из подходов не решает напрямую проблему построения метрики качества потока данных, который мог быть информативным для отслеживания «здоровья» потока данных.

Пробелы и открытые вопросы

Отсутствие унифицированного механизма метрики качества потока данных без разметки. Существующие подходы либо выводят латентное представление, либо возвращают суррогатные функции потерь, но не предоставляют нормированного показателя, сопоставимого, который можно сравнить между потоками;

Онлайн-адаптация при дрейфе концепта. Большинство методов обучаются офлайн; попытки инкрементального изменения параметров (например, EMA-обновление в BYOL) недостаточно изучены на потоках реального времени;

Интерпретируемость. Для инженерных команд важно понимать, какой признак привёл к срабатыванию. В контрастивных методах метрика — это скрытый косинус/ранг, который сложно явно интерпретировать без дополнительных средств обработки.

Тем самым формируется задача: объединить ранговый самообучающийся метод с легковесным онлайн-энкодером и вывести универсальную метрику качества потока данных, пригодный для потоковой обработки, автоматически адаптирующийся к дрейфу и интерпретируемый через важность признаков.

Постановка задачи

Рассмотрим бесконечный поток событий

$$S = \{x_t \in \mathbb{R}^{d_t} t \in \mathbb{N}\}, \quad (1)$$

поступающий из шины Apache Kafka и обрабатываемый с гарантией обработки строго один раз, даже в случае сбоев, перезапусков или повторной передачи данных. Мы называем такой поток неизученным, если перед моментом t=0 отсутствуют:

- декларативное описание схемы данных;
- эталонный набор данных или эталонный набор правил качества данных;
- историческая разметка истинности данных.

Пусть скользящее окно длиной w представлено формулой:

$$W_t = \{x_{t-w+1}, \ldots, x_t\}, \quad (2)$$

Требуется вычислять:

$$DQS(t) = g\left(\{s_\theta(x_i)\}_{x_i \in W_t}\right) \in [0,1], \quad (3)$$

где

онлайновость — g обновляется инкрементно, без перерасчёта всего временного окна;

нормированность — диапазон [0,1] обеспечивает пороговую интерпретацию;

монотонность по качеству — если две подпоследовательности совпадают;

сравнимость между потоками при различных схемах.

Определим оптимизация ранжирующей функции с псевдо-сигналом качества. Для каждого окна W_t формируем набор пар:

$$P_t = \{(x_i, x_j, y_{ij} = 1)\} \cup \{(x_i, x_k, y_{ik} = 0)\}, \quad (4)$$

где положительные примеры получаются слабыми аугментациями оригинального события, а отрицательные примеры получаются сильными трансформациями либо дальними во-времени точками, что эмпирически имитирует «искажённое» состояние данных [16–17].

Искомый скорер обучается минимизацией ранговой функции потерь:

$$\mathcal{L}_t(\theta) = \frac{1}{|P_t|} \sum_{(u,v,y) \in P_t} \begin{bmatrix} y\sigma(1 - s_\theta(u) + s_\theta(v)) + \\ +(1-y)\sigma(1 + s_\theta(u) - s_\theta(v)) \end{bmatrix}, \quad (5)$$

где σ — гладкая аппроксимация шарнирная функция потерь.

Финальный показатель качества определяется как:

$$DQS(t) = \frac{1}{w} \sum_{x_i \in W_t} \Phi\left(s_\theta(x_i)\right), \quad (6)$$

где Φ — сигмоидальный коэффициент масштабируемости.

Тем самым показатель качества потоковых данных — это агрегированный относительный ранг, инвариантный к абсолютным шкалам признаков.

Таким образом, формализована задача самообучаемой генерации метрики качества для неизученного потока с ограничениями в режиме реального времени, сводимая к оптимизации ранговой функции в скользящем.

Предлагаемый метод самообучающегося ранжирования

Предлагаемая система метрики качества данных построена по принципу микро-сервиса, на логическом состоит из четырёх взаимосвязанных модулей:

Онлайн-энкодер. Лёгкая 1-D архитектура Temporal Causal CNN + GRN кодирует каждый поступивший элемент $x_t \in R^{d_t}$ в латентное пространство $z_t \in R^h$ [18].

Модуль маскирующей аугментация. Для каждого x_t генерируется серия аугментированных копий [19].

Модуль парного сравнения. Параллельный двуслойный MLP со общими весами, вычисляющий скалярные скореры. На его выходе реализуется ранговая функция — ядро потерь оптимизируется по принципу RankSim [20].

Модуль агрегации. Инкрементальный подсчёт метрики качества данных выполняется на основе относительных рангов в скользящем окне W_t.

Подобная модульность упрощает замену энкодера или функции. Сформируем комплексной показатель качества потока данных. Скореры $s_\phi(x_i)$ отражают относительное доверие к отдельным элементам. Для перехода к метрике используется:

$$DQS(t) = \frac{1}{w} \sum_{x_i \in W_t} \sigma\left(\beta\left[s_\phi(x_i) - \mu_t\right]\right), \quad (7)$$

где μ_t — медиана скореров в окне W_t, β — коэффициент «контрастности». Такая нормировка обеспечивает:

Инвариантность к смещению: добавление константы не изменяет показатель качества потока данных.

Сравнимость потоков: значение всегда лежит в [0,1] благодаря сигмоиде.

Чувствительность к хвостам: при появлении кластера аномалий хвост распределения смещается, что резко уменьшает интегральный балл.

Экспериментальная часть

Для проверки работоспособности метода взят открытый поток NYC Taxi Trip Record 2019–2020–143 млн событий о поездках такси с восемью основными полями (время, координаты, стоимость и др.) [21]. Логи были проиграны в Kafka с ускорением в десять раз, что дало стабильную нагрузку порядка ста тысяч сообщений в секунду.

Сравниваемые решения

Для сравнения мы использовали Rule-based (GX) [22] — это метод, в котором модель обучается извлекать представления, основываясь на предопределённых, интерпретируемых правилах, описывающих закономерности или шаблоны поведения во временных данных; Isolation Forest [23] — это метод машинного обучения для обнаружения аномалий, основанный на изолировании выбросов с помощью случайных деревьев; Deep SVDD (Support Vector Data Description) [24] — метод обнаружения аномалий с использованием глубоких нейросетей, который обучает модель так, чтобы все нормальные данные сжимались внутрь компактной области в скрытом признаковом пространстве. В таблице 1 представлено краткое описание методов.

Таблица 1.

Методы для сравнения

Метод	Краткое описание
Предложенный метод	Предлагаемый самообучающийся ранжирующий метод
Rule-based (GX)	12 ручных проверок Great Expectations
Isolation Forest	100 деревьев, окно = 1024
Deep SVDD	Одноклассовая CNN-модель, дообучение каждые 10 000 записей

Сравнение проходило по трём метрикам. Точность P1000 — доля реальных ошибок среди первых 1000 срабатываний. Задержка обнаружения — среднее время между вводом искусственной ошибки и срабатыванием. Ресурсы — средняя загрузка процессора. В таблице 2 представлены значения метрик по каждому из методов.

Предложенный метод точнее rule-based на +31 п.п. и реагирует на ошибки почти в шесть раз быстрее. При этом он легче Deep SVDD по CPU в три. Бутстрап-тест подтвердил значимость разницы с Isolation Forest ($p < 0{,}01$).

Таблица 2.

Результаты сравнения

Метод	P1000	Задержка, с	CPU, vCPU
Rule-based	0,43	184	0,18
Isolation Forest	0,59	77	0,91
Deep SVDD	0,68	60	1,83
Предложенный метод	0,74	32	0,55

Преимущества метода

Полная автономия от меток и схемы. Метрика качества данных формируется исключительно из входного потока, устраняя необходимость в эталонных данных или ручных правилах. Сочетание превентивного и реактивного контроля. Аугментированные положительные примеры имитируют погрешности сенсоров, а отрицательные примеры — тяжёлые искажения, поэтому ранговый скорер обнаруживает как мелкие, так и катастрофические нарушения качества быстрее проверок, основанных на правилах. Интерпретируемость через важность признаков. Метод позволяет автоматически строить отчёты о важности параметров, что упрощает работу без глубокого экспертного опыта в машинном обучении у пользователя.

Выводы

Предложен метод оценки качества потока данных, объединяющий самообучающийся ранжирующий энкодер, скользящее агрегирование и встраивание в потоки данных. Метод может работать без каких-либо разметок данных и схемы — это закрывает существенный пробел в существующих фреймворках качества данных.

Заключение

Работа демонстрирует, что самообучающееся ранжирование является жизнеспособным и экономичным решением для автоматического контроля данных в динамичных потоковых системах. Полученные результаты открывают дорогу к унифицированной «сквозной» метрике качества, способной эволюционировать вместе с данными без участия человека.

ЛИТЕРАТУРА

1. Waehner K. Top 5 Trends for Data Streaming with Kafka and Flink in 2024 [Электронный ресурс]. — URL: https://www.kai-waehner.de/blog/2023/12/02/top-5-trends-for-data-streaming-with-apache-kafka-and-flink-in-2024/ (дата обращения: 11.05.2025).
2. Data Dynamics. Unveiling the Shadows: Dark Data's Menacing Impact on Security and Compliance in the Age of AI [Электронный ресурс]. — URL: https://www.datadynamicsinc.com/blog-unveiling-the-shadows-dark-datas-menacing-impact-on-security-and-compliance-in-the-age-of-ai/ (дата обращения: 11.05.2025).
3. CNCF. Observability Trends in 2025 — What's Driving Change? [Электронный ресурс]. — URL: https://www.cncf.io/blog/2025/03/05/observability-trends-in-2025-whats-driving-change/ (дата обращения: 11.05.2025).
4. Gröger F. и др. Intrinsic Self-Supervision for Data Quality Audits // Advances in Neural Information Processing Systems 37. — 2024. — DOI: 10.48550/arXiv.2305.17048.
5. Cao L. и др. Breaking Annotation Barriers: Generalized Video Quality Assessment via Ranking-Based Self-Supervision [Электронный ресурс]. — arXiv:2505.03631, 2025.
6. Great Expectations. Streaming data support in Great Expectations [Электронный ресурс]. — URL: https://discourse.greatexpectations.io/t/streaming-data-support-in-great-expectations/389 (дата обращения: 11.05.2025).
7. FirstEigen. Overcome the Limitations of AWS Deequ with AI/ML [Электронный ресурс]. — URL: https://firsteigen.com/aws-deequ-lp/ (дата обращения: 11.05.2025).
8. Amazon Web Services. Test data quality at scale with Deequ // AWS Big Data Blog. — 16 июня 2018. — URL: https://aws.amazon.com/blogs/big-data/test-data-quality-at-scale-with-deequ/ (дата обращения: 11.05.2025).
9. Evidently AI. Model monitoring for ML in production: a comprehensive guide [Электронный ресурс]. — URL: https://www.evidentlyai.com/ml-in-production/model-monitoring (дата обращения: 11.05.2025).
10. Evidently AI. Machine Learning Monitoring, Part 3: What Can Go Wrong With Your Data [Электронный ресурс]. — URL: https://www.evidentlyai.com/blog/machine-learning-monitoring-what-can-go-wrong-with-your-data (дата обращения: 11.05.2025).
11. Eldele E. и др. Time-Series Representation Learning via Temporal and Contextual Contrasting (TS-TCC) // Proc. IJCAI-21. — 2021.
12. Zhang X., Wu Y. Self-Supervised Learning for Time Series: Contrastive or Generative? // arXiv:2403.09809, 2024.
13. Yang P. и др. RankSim: Ranking Similarity Regularization for Deep Imbalanced Regression // arXiv:2205.15236, 2022.
14. Garrido Q. и др. RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank // Proc. ICML 2023.
15. Wu Y. и др. Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency // Proc. NeurIPS 2022.
16. Wu Y. et al. Generalized Video Quality Assessment via Ranking-Based Self-Supervision // arXiv:2505.03631, 2025.
17. Gröger F. et al. Intrinsic Self-Supervision for Data Quality Audits // Advances in Neural Information Processing Systems 37, 2024.
18. Grill J.-B. и др. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning [Электронный ресурс]. — arXiv:2006.07733, 2020. — URL: https://arxiv.org/abs/2006.07733 (дата обращения: 11.05.2025).
19. Tschannen M., Jain M. Time Series Augmentations for Self-Supervised Representation Learning [Электронный ресурс]. — arXiv:2210.06824, 2022.
20. Yang P. и др. RankSim: Ranking Similarity Regularization for Deep Imbalanced Regression [Электронный ресурс]. — arXiv:2205.15236, 2022.
21. New York City Taxi & Limousine Commission. TLC Trip Record Data 2019–2020 [Электронный ресурс]. — URL: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page (дата обращения: 11.05.2025).
22. De Bridgne C. Practical Data Quality with Great Expectations. — O'Reilly Media, 2023.
23. Liu F.T., Ting K.M., Zhou Z.-H. Isolation Forest // IEEE ICDM '08. — Pisa, 2008.
24. Ruff L. и др. Deep One-Class Classification // Proc. ICML 2018. — 2018.

Журнал «Современная наука: актуальные проблемы теории и практики»