

УДК 004.4
DOI 10.17513/snt.40302

ВНЕДРЕНИЕ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ В ПОТОКОВОЙ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ СИСТЕМЕ

¹Ермаков С.Р., ²Зыков С.В.

¹ФГБОУ ВО «МИРЭА – Российский технологический университет»,
Москва, e-mail: ermakov_s@mirea.ru;

²ФГАОУ ВО «Национальный исследовательский университет
„Высшая школа экономики“», Москва, e-mail: zykov@hse.ru

Цель исследования заключается в разработке нового архитектурного решения, обеспечивающего внедрение моделей потокового машинного обучения на основе методов и алгоритмов потоковой обработки многомодальных данных и машинного обучения в потоковой интеллектуальной образовательной системе для повышения персонализации образовательного процесса и автоматической адаптации учебных материалов под уровень знаний обучающихся в реальном времени. В работе описаны процессы передачи и обработки данных из различных многомодальных источников в модель потокового машинного обучения. Для нормализации данных применяется адаптивный метод скользящих окон, позволяющий корректно обрабатывать непрерывный поток информации с учетом временной синхронизации. Особое внимание в исследовании уделено вопросу контроля использования оперативной памяти для обеспечения стабильного функционирования системы в условиях ограниченности вычислительных ресурсов в образовательных организациях. Экспериментальные результаты, визуализированные на графиках и в таблице, демонстрируют эффективность интеграции нескольких алгоритмов потокового машинного обучения, таких как адаптивный случайный лес, инкрементальный алгоритм дерева решений для задач классификации и регрессии, в единую модель. Предложенная архитектура, которая сопровождается схемой, характеризуется высокой масштабируемостью, легкостью интеграции с существующими образовательными платформами и адаптивностью к изменяющимся условиям использования, что открывает перспективы дальнейших исследований и практического применения в образовательных технологиях.

Ключевые слова: потоковое машинное обучение, потоковые данные, интеллектуальная образовательная система, регрессия, классификация

DEPLOYMENT AND OPERATION OF MACHINE LEARNING MODELS IN A STREAMING INTELLIGENT EDUCATIONAL SYSTEM

¹Ermakov S.R., ²Zykov S.V.

¹MIREA – Russian Technological University, Moscow, e-mail: ermakov_s@mirea.ru;

²National Research University Higher School of Economics, Moscow, e-mail: zykov@hse.ru.

The purpose of the research is to develop a new architectural solution that ensures the implementation of streaming machine learning models based on methods and algorithms for streaming multimodal data processing and machine learning in a streaming intelligent educational system to enhance the personalization of the educational process and automatically adapt educational materials to the level of knowledge of students in real time. The paper describes the processes of data transmission and processing from various multimodal sources into a streaming machine learning model. To normalize the data, the adaptive sliding window method is used, which allows for the correct processing of a continuous flow of information, considering time synchronization. Special attention in the study is paid to the issue of controlling the use of RAM to ensure the stable functioning of the system in conditions of limited computing resources in educational institutions. The experimental results, visualized on graphs and in a table, demonstrate the effectiveness of integrating several streaming machine learning algorithms, such as adaptive random forest, an incremental decision tree algorithm for classification and regression problems into a single model. The proposed architecture, which is accompanied by the scheme, is characterized by high scalability, ease of integration with existing educational platforms and adaptability to changing conditions of use, which opens prospects for further research and practical application in educational technologies.

Keywords: streaming machine learning, streaming data, intelligent educational system, regression, classification

Введение

Сегодня перед современными институтами образования стоит задача повышения персонализации и эффективности обучения. Технический аспект решения этой задачи сталкивается с несколькими проблемами, таким как: острый недостаток технологических решений, программного обеспечения, которые могли бы интегрировать

интеллектуальные подходы в электронные обучающие системы (ЭОС); ограниченность вычислительных ресурсов в образовательных организациях (ОО). В основном, сейчас в ОО используются ЭОС без интеллектуальных подходов к индивидуализации обучения и регулированию использования вычислительных ресурсов [1, 2]. Авторами предлагается новая концепция *потоковой*

интеллектуальной образовательной системы (ПИОС). Она представляет собой следующий этап развития ЭОС в ОО и основана, в том числе, на реализации моделей, методов и алгоритмов машинного обучения и анализа данных обучающихся с контролем использования вычислительных ресурсов (например, оперативной памяти). В данном исследовании авторы описывают практическое внедрение моделей машинного обучения в ПИОС и их применение с использованием модифицированных и разработанных методов и алгоритмов потоковой обработки данных и потокового машинного обучения.

Цель исследования – разработка нового архитектурного решения, обеспечивающего внедрение моделей потокового машинного обучения на основе методов и алгоритмов потоковой обработки многомодальных данных и машинного обучения в потоковой интеллектуальной образовательной системе для повышения персонализации образовательного процесса и автоматической адаптации учебных материалов к уровню знаний обучающихся в реальном времени.

Материалы и методы исследования

В ходе исследования был использован набор программных инструментов с открытым исходным кодом: система управления электронными обучающими курсами Moodle, платформа для потоковой обработки и мониторинга данных Apache Kafka, Python с библиотекой для потокового машинного обучения River [3], платформа для сбора метрик Prometheus, приложение для визуализации данных Grafana [4]. Эксперименты проводили на гетерогенной вычислительной платформе с центральным процессором (CPU) Intel® Core™ i7-8700 CPU, 3.20GHz (6 физических ядер, 12 потоков), видеокартой (GPU) NVIDIA GeForce RTX 2070, 1410.0 MHz, 8 GB GDDR6. Данные для экспериментальной оценки предложенных решений были взяты из датасета Junyi Academy Online Learning Activity Dataset [5], также были симулированы и обработаны ЭЭГ-сигналы с помощью библиотеки NeuroKit2 для Python [6].

Для подготовки предметного поля исследования был проведен анализ описанных в русскоязычной и зарубежной научной литературе: существующих методов и технологий в области потоковой обработки данных, подходов к реализации архитектур интеллектуальных обучающих и образовательных систем, модулей и сервисов в них, подходов к использованию нейроинтерфейсов в образовательных системах. Формализация процесса потоковой передачи и обработки данных проведена с использованием

методов математического моделирования. Модели машинного обучения были реализованы с применением модифицированных и разработанных методов и алгоритмов потокового машинного обучения. Для описания архитектурного решения авторами применены методы системного анализа и проектирования. Экспериментальная оценка архитектурного решения проведена с помощью симуляции потоков данных из описанных наборов данных. Верификация результатов исследования осуществлена с помощью оценки метрик, записанных программными инструментами.

В настоящее время в образовательных учреждениях широко применяются ЭОС, такие как Moodle, Blackboard, Stepik и др. Эти платформы обеспечивают доступ к учебным материалам, управление курсами, проведение тестирований и мониторинг прогресса обучающихся. Однако большинство из них используют статическую модель взаимодействия, в которой анализ данных осуществляется постфактум, а адаптация учебного контента не происходит в реальном времени. Это ограничивает возможности персонализированного обучения и адаптивных образовательных стратегий. Одним из ключевых недостатков традиционных ЭОС является их неспособность эффективно обрабатывать потоковые данные в режиме реального времени. Как правило, сейчас платформы предлагают инструменты для анализа учебной активности студентов, но эти механизмы основаны на пакетной обработке данных, что затрудняет динамическую подстройку контента под индивидуальные потребности студентов [7, 8].

В отличие от ЭОС, предлагаемая авторами потоковая интеллектуальная образовательная система (ПИОС) решает эти проблемы за счет интеграции моделей потокового машинного обучения, которые позволяют анализировать поступающие данные в реальном времени. Благодаря этому система может моментально адаптировать содержание учебных курсов, подстраивать уровень сложности заданий и предоставлять преподавателям оперативные аналитические отчеты. Например, если студент испытывает затруднения с определенной темой, ПИОС может автоматически предложить дополнительные материалы или альтернативные методики обучения, повышая эффективность образовательного процесса.

Таким образом, ПИОС представляет собой следующий этап эволюции электронных обучающих систем, сочетая в себе принципы потоковой обработки данных и интеллектуального анализа учебной активности. Это позволяет значительно повы-

сить адаптивность и персонализацию образовательных процессов, что особенно актуально для создания современных цифровых образовательных сред.

В ЭОС существует множество различных потоков данных, которые могут быть использованы для построения интеллектуальных систем. Для экспериментального исследования ПИОС авторами были выделены и проанализированы потоки данных, указанные в таблице 1.

Для эффективного анализа таких данных в режиме реального времени необходимо архитектурное решение, которое даст возможность быстрой адаптации образовательного контента и стратегии преподавания. Например, если система обнаруживает, что студент тратит на задание значительно больше времени, чем другие учащиеся, она может автоматически предложить дополнительные разъяснения или скорректировать сложность следующего вопроса. Кроме того, использование потокового анализа данных позволяет оптимизировать нагрузку на серверные мощности образовательных платформ, так как информация обрабатывается поступательно, без необходимости хранения и анализа больших объемов данных в виде статических файлов. Это особенно важно для образовательных учрежде-

ний, использующих облачные платформы и дистанционные технологии обучения, где вычислительные ресурсы ограничены.

Применение моделей потокового машинного обучения в современных интеллектуальных образовательных системах может не только повысить точность прогнозирования учебных результатов, но и способствовать созданию персонализированной среды обучения, адаптированной под каждого обучающегося в реальном времени. Для обеспечения эффективной работы ПИОС учитываются параметры потоков данных, которые влияют на выбор методов и алгоритмов обработки.

1. Частота обновления данных: например, события активности студентов (логины, клики, работа с контентом) поступают каждую секунду, тогда как оценки тестов обновляются реже.

2. Размер данных: потоковые данные из ЭОС содержат короткие текстовые записи, в то время как нейроинтерфейсы и видеопотоки могут генерировать гигабайты информации в минуту.

3. Требования к реальному времени: прогнозирование успеваемости студента может выполняться раз в сутки, а адаптация учебного материала должна происходить в момент взаимодействия с системой.

Таблица 1

Потоки данных в ЭОС

Источник данных	Описание потока данных	Варианты использования	Частота обновления
Действия в ЭОС (Moodle, Blackboard, Stepik и др.)	Время входа, посещенные страницы, просмотренные материалы, длительность просмотра, количество попыток в тестах	Определение вовлеченности, предложение дополнительных материалов, прогнозирование успеваемости	В реальном времени
Результаты тестов и заданий	Оценки, количество попыток, время выполнения заданий, количество ошибок	Автоматическая корректировка сложности заданий, выдача адаптивных рекомендаций	После каждого теста
История взаимодействия с системой	История нажатий, навигация по курсам, поиск по материалам	Определение интересов студента, персонализация содержания курсов	В реальном времени
Анализ текстов в общих форумах	Вопросы студентов, ответы преподавателей, частота сообщений, эмоциональная окраска текста	Рекомендации на основе тем обсуждений, выявление трудностей у студентов	В реальном времени или раз в сутки
Прохождение видеоуроков	Досмотрено ли видео до конца, количество перемоток, скорость воспроизведения	Определение сложных тем, предложение поясняющих материалов	В реальном времени
Биометрические данные (при наличии нейроинтерфейсов)	Электроэнцефалограмма (ЭЭГ), уровень стресса, сердечный ритм	Регулировка сложности заданий: при высоком стрессе – упрощение, при высокой концентрации – усложнение	В реальном времени
Опросы и обратная связь студентов	Оценка сложности курса, предпочтительные методы обучения, отзывы	Улучшение курсов, персонализация методик обучения	Раз в неделю или по завершении блока
Исторические данные о студенте	Прошлые курсы, успеваемость в других дисциплинах, стиль обучения	Рекомендация материалов на основе предыдущего опыта обучения	Перед началом курса

В результате в ПИОС используются разные стратегии обработки: низкочастотные данные (оценки, тесты) анализируются периодически, применяются для стратегических прогнозов; высокочастотные потоки (активность, сигналы) обрабатываются непрерывно, обеспечивая адаптацию контента в режиме реального времени.

Предположим, что студент выполняет задание в ЭОС (например, тест в Moodle). Поток данных может выглядеть следующим образом:

1. Студент начинает попытку → фиксируется логин в систему, начало теста.

2. Фиксируются действия в процессе тестирования → время на вопрос, количество изменений ответа.

3. Обрабатывается контекстная информация → данные с устройств ввода, физиологические показатели (если используются нейроинтерфейсы).

4. Система анализирует сложность вопросов в режиме реального времени → если студент долго задерживается на одном вопросе, ПИОС может предложить подсказку или изменить алгоритм выдачи заданий.

5. После завершения теста данные агрегируются → оценивается уровень усвоения материала, обновляется персонализированная траектория обучения.

Этот процесс отличается от ЭОС, где анализ выполняется постфактум, а не в момент взаимодействия студента с системой. Внедрение потокового машинного обучения в образовательные системы позволит обрабатывать разнородные данные из ЭОС, сенсоров, видеопотоков и тестов в режиме реального времени.

Формализуем процесс потоковой передачи данных, получаемых из многомодальных источников (например, из Moodle, нейроинтерфейсов, других устройств и датчиков ввода информации через API), с акцентом на их синхронизацию и адаптивную нормализацию. Основной задачей на данном этапе является интеграция разнородных данных, поступающих в реальном времени, в единую потоковую систему, что может позволить эффективно использовать их для обучения моделей потокового машинного обучения. Пусть имеется K различных источников данных, где каждый источник представляет поток данных с собственными характеристиками и модальностью. Обозначим эти источники как (1):

$$S^{(k)} = \left\{ s^{(k)}(t^{(k)}) \mid t^{(k)} \in T^{(k)} \right\}, \quad (1)$$

$$k = 1, 2, \dots, K,$$

где $s^{(k)}(t^{(k)})$ – элемент потока k -го источника, а $t^{(k)}$ – временная метка, ассоциированная

с этим источником. Для обеспечения корректной агрегации данных требуется синхронизация временных меток, что может быть достигнуто использованием общего времени (например, через протокол сетевого времени) или алгоритмов выравнивания временных рядов.

Каждый поток данных $s^{(k)}(t)$ подвергается индивидуальной программной обработке в программном компоненте Apache Kafka Producer с целью приведения к единой шкале и устранения выбросов. Если для каждого источника используется адаптивное масштабирование с экспоненциальным сглаживанием [9], то обновление среднего значения (2) и дисперсии (3) можно записать следующим образом:

$$\mu_t^{(k)} = \mu_{t-1}^{(k)} + \alpha \cdot (s^{(k)}(t) - \mu_{t-1}^{(k)}), \quad (2)$$

$$\left(\sigma_t^{(k)} \right)^2 = (1 - \alpha) \cdot \left(\sigma_{t-1}^{(k)} \right)^2 + \alpha \cdot (s^{(k)}(t) - \mu_{t-1}^{(k)})^2, \quad (3)$$

где α – коэффициент сглаживания, одинаковый или адаптированный для каждого источника.

Параллельно для данных, поступающих из каждого источника, рассчитываются выборочные статистики на основе скользящего окна фиксированной длины W с применением метода ADWIN [10]. Таким образом, нормализованное значение для источника k определяется по формуле (4):

$$z_t^{(k)} = \frac{s^{(k)}(t) - \bar{s}_w^{(k)}}{\sigma_w^{(k)}}, \quad (4)$$

где $\bar{s}_w^{(k)}$ и $\sigma_w^{(k)}$ – выборочное среднее и стандартное отклонение, вычисленные по данным, попавшим в окно W для источника k .

Для объединения информации из всех K источников используется оконная агрегация. Если назначить общее окно времени $[\tau, \tau + \Delta t]$, то агрегированный вектор данных можно представить как (5):

$$x(\tau) = g \left(\left\{ z_t^{(k)} \mid t \in [\tau, \tau + \Delta t], k = 1, \dots, K \right\} \right), \quad (5)$$

где функция $g(\cdot)$ осуществляет объединение данных по временным окнам, учитывая особенности каждой модальности. В результате формируется единый вектор $x(\tau)$, характеризующий данные, полученные за интервал времени, из всех источников.

Передача агрегированных данных осуществляется через потоковую систему обмена сообщениями из Apache Kafka Producer. Каждое сообщение, содержащее $x(\tau)$ с соответствующей временной меткой, направляется в категории (topics) кластера (Apache Kafka Cluster), которые далее используются для обучения модели.

Для решения задач классификации и регрессии ПИОС в программном компоненте Apache Kafka Consumer, где реализуется модель потокового машинного обучения, используется комбинированный метод обучения [11]. Совокупная функция потерь описывается как (6):

$$L(f) = \alpha \cdot L_c(f_c) + (1 - \alpha) \cdot L_r(f_r), \quad (6)$$

где $L_c(f_c)$ – функция потерь для задачи классификации, кросс-энтропия (7):

$$L_c(f_c) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C 1[y_c^{(i)} = j] \cdot \log p_j^{(i)}, \quad (7)$$

где $L_r(f_r)$ – функция потерь для задачи регрессии, среднеквадратичная ошибка (8):

$$L_r(f_r) = \frac{1}{N} \sum_{i=1}^N (y_r^{(i)} - \hat{y}_r^{(i)})^2, \quad (8)$$

коэффициент $\alpha \in [0,1]$ регулирует вклад каждой задачи. Обновление параметров модели осуществляется в потоковом режиме, например по правилу, аналогичному стохастическому градиентному спуску (9):

$$\theta(\tau+1) = \theta(\tau) - \eta \cdot \nabla_{\theta} L(\theta(\tau); x(\tau), y(\tau)), \quad (9)$$

где $y(\tau)$ включает как классовые метки, так и регрессионные значения, а η – параметр скорости обучения. В реализации комбинированной модели машинного обучения можно перекрестно использовать адаптивный случайный лес и инкрементальное дерево решений для задач классификации и регрессии [12].

Метрики (metrics) производительности – как всего Apache Kafka Cluster, так и Consumer с моделью машинного обучения – считываются Prometheus и визуализи-

руются с помощью Grafana. В качестве общего формата передачи сообщений между программными компонентами используется JSON. Предсказанные моделью признаки передаются в образовательный модуль ПИОС (например, обратно в Moodle), где на их основе для обучающихся и преподавателей формируются персонализированные рекомендации.

Результаты исследования и их обсуждение

Представленный процесс можно реализовать в виде архитектурного решения для внедрения потоковых моделей машинного обучения в ПИОС. Такое решение обеспечивает интеграцию многомодальных источников данных, позволяя одновременно учитывать особенности каждой модальности, проводить адаптивную нормализацию и синхронизацию, а также использовать комбинированные методы обучения для решения задач классификации и регрессии в режиме реального времени. Схема предлагаемого архитектурного решения представлена на рисунке 1.

В рамках исследования в Moodle и Apache Kafka Producer был использован набор данных Junyi Academy, предоставленный некоммерческой организацией, занимающейся онлайн-обучением. Этот датасет содержит данные о взаимодействии обучающихся с образовательной платформой, в том числе временные метки активности, результаты тестов и попытки прохождения заданий. Выбор датасета был обусловлен тем, что он предоставляет структурированные временные ряды образовательной активности, что позволяет технически настроить и откалибровать модели машинного обучения в контролируемой экспериментальной среде.

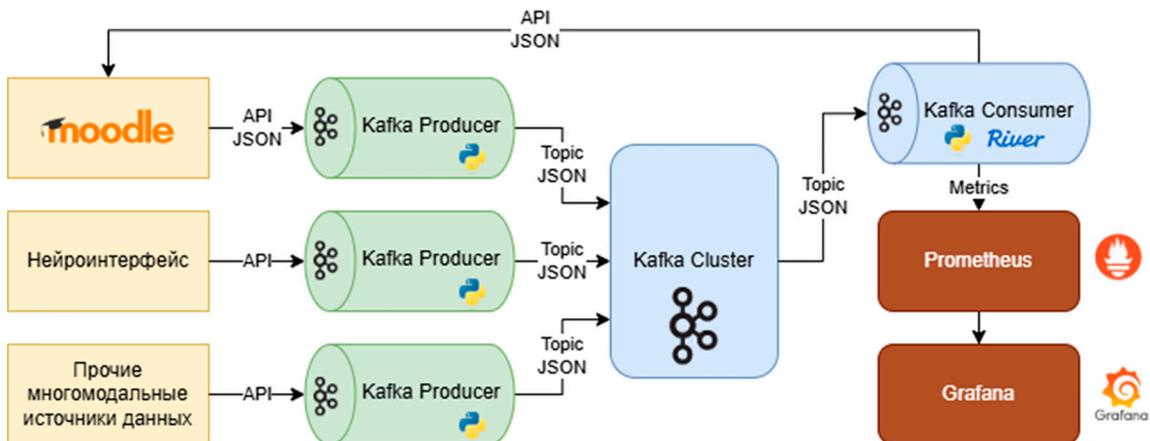


Рис. 1. Предлагаемое архитектурное решение для внедрения моделей машинного обучения в ПИОС

```

# Основной класс для обработки потоковых данных
class StreamingMLModule:
    def __init__(self, kafka_server, topic, models_config, metrics_file="metrics.csv"):
        """
        Модуль для обработки потоковых данных с использованием Kafka и обучения нескольких моделей.

        Параметры:
        kafka_server: Адрес Kafka сервера
        topic: Название топика Kafka
        models_config: Конфигурация моделей
        metrics_file: Путь к файлу для записи метрик
        """
        # Настройка KafkaConsumer для получения сообщений
        self.consumer = KafkaConsumer(
            topic,
            bootstrap_servers=kafka_server,
            value_deserializer=lambda m: json.loads(m.decode('utf-8')),
            auto_offset_reset='earliest',
            enable_auto_commit=True
        )
        # Инициализация моделей
        self.models = self.initialize_models(models_config)
        # Хранение информации об использовании памяти
        self.memory_usage = {name: [] for name in self.models}
        self.iterations = []
        # Метрики для классификации и регрессии
        self.classification_metrics = {
            name: LogLoss() for name in self.models if 'Classifier' in name or 'Combined' in name
        }
        self.regression_metrics = {
            name: MSE() for name in self.models if 'Regressor' in name or 'Combined' in name
        }
        self.metrics_file = metrics_file

```

Рис. 2. Фрагмент кода реализации комбинированной модели

Использование набора данных дало возможность оценить работоспособность алгоритмов потокового машинного обучения, протестировать их адаптивность к изменяющимся условиям и оптимизировать параметры модели перед развертыванием в реальных образовательных средах. Несмотря на возможные различия в подходах в разных образовательных системах, ключевые закономерности в данных являются универсальными, что делает результаты эксперимента применимыми к различным образовательным системам.

В Apache Kafka Consumer были реализованы и сравнены пять моделей потокового машинного обучения для решения задачи регрессии: три на основе алгоритма инкрементального дерева решений с применением неравенства Хефффдинга (Hoeffding Tree): HT1 и HT2 без механизма адаптации к концептуальному дрейфу и НАТ с таким механизмом, потоковое градиентное дерево (Streaming Gradient Tree, SGT), комбинированная модель с применением метода масштабирования признаков

и скользящего окна (EMA & SW Model, EMAWM). Последняя модель создана на основе НАТ – алгоритма адаптивного случайного леса – и дополнена авторскими разработками: методом масштабирования признаков, а также параметрами ограничения памяти и учета давности данных через коэффициент затухания. Такая модель позволяет эффективно использовать оперативную память, удаляя устаревшие узлы, что делает ее подходящей для работы в условиях ограниченных ресурсов. Фрагмент кода реализации комбинированной модели представлен на рисунке 2.

Для оценки качества предсказаний моделей использовались стандартные метрики: *среднеквадратичная ошибка (MSE)*, измеряющая среднеквадратичное отклонение предсказанных значений от истинных, и *коэффициент детерминации (R^2)*, отражающий долю дисперсии зависимой переменной, объясненной моделью. Обучение каждой модели проводилось итеративно, с обновлением параметров после обработки каждого нового примера из потока.

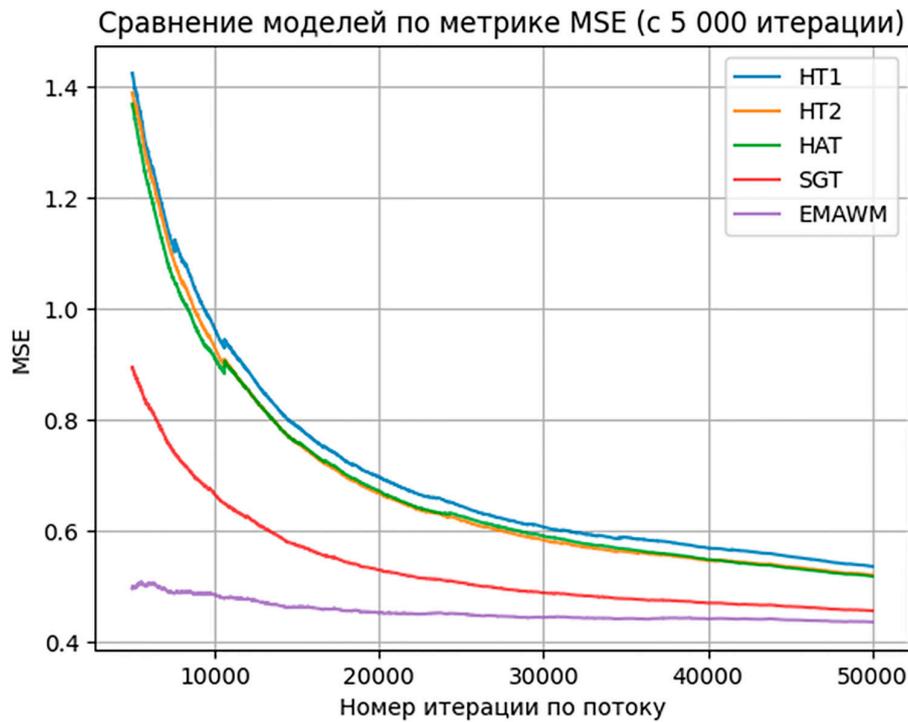


Рис. 3. Результаты эксперимента по метрике MSE

Таблица 2

Результаты измерения MSE и R²

Наблюдение	Метрика	HT1	HT2	HAT	SGT	EMAWM
10000	MSE	0.96	0.93	0.91	0.67	0.48
	R ²	-0.01	0.03	0.04	0.30	0.49
20000	MSE	0.70	0.67	0.67	0.53	0.45
	R ²	0.27	0.30	0.30	0.45	0.53
30000	MSE	0.61	0.58	0.59	0.49	0.44
	R ²	0.36	0.39	0.38	0.49	0.53
40000	MSE	0.57	0.55	0.55	0.47	0.44
	R ²	0.40	0.43	0.43	0.51	0.54
50000	MSE	0.54	0.52	0.52	0.46	0.44
	R ²	0.44	0.46	0.46	0.52	0.54

На каждой итерации выполнялось предсказание целевой переменной (время в минутах, затраченное на попытку решения примера в тесте в Moodle) с соответствующими признаками: данными активности в ЭОС и симулированными сигналами нейроинтерфейса. Каждой модели было подано 50 000 наблюдений. Диапазон значений целевой переменной от 0 до 5,56 указывает на наличие как низких, так и высоких значений, что может свидетельствовать о разнообразии примеров в наборе данных. Дисперсия целевой переменной

рассчитывается как квадрат стандартного отклонения $\sigma^2 = \text{std}^2 \approx 0.977725^2 \approx 0.955$ [13]. График, отображающий значения среднеквадратичной ошибки, представлен на рисунке 3.

Значение MSE для модели EMAWM ($\text{MSE}_{\text{EMAWM}} = 0.43$) более чем в два раза ниже дисперсии целевой переменной. Базовая модель, которая всегда предсказывает среднее значение целевой переменной, будет иметь MSE, равный дисперсии данных. Следовательно, $\text{MSE}_{\text{EMAWM}}$ показывает удовлетворительные результаты для применения в ПИОС.

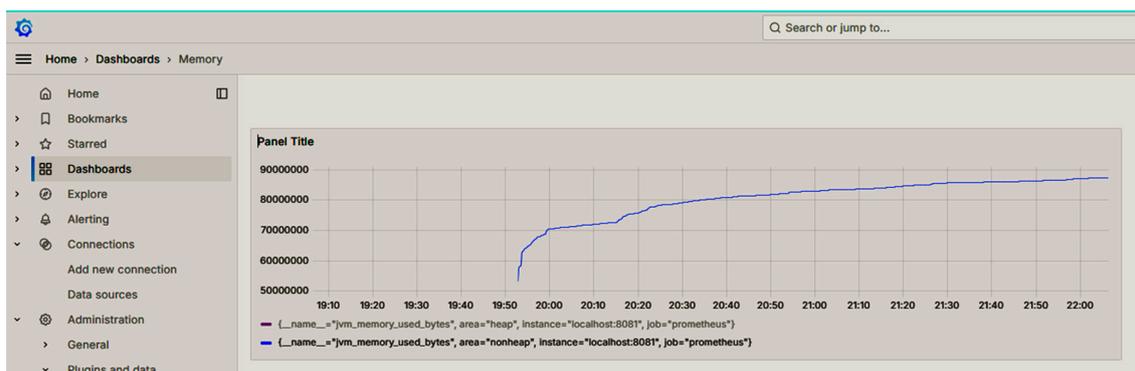


Рис. 4. Визуализация использования памяти моделями потокового машинного обучения в Grafana

Экспериментальное вычисление метрики R^2 дало возможность убедиться в отсутствии ошибок в расчете метрики MSE и подтвердить вывод об эффективности работы модели потокового машинного обучения с масштабированием данных. Результаты измерения метрик по всем исследуемым моделям отражены в таблице 2.

Эти метрики наряду с измерениями производительности всего Kafka Cluster систематически передаются в систему мониторинга Prometheus, что позволяет в реальном времени отслеживать как ключевые показатели качества моделей, так и характеристики инфраструктуры. Собранные данные затем визуализируются с помощью Grafana, где представляются наглядные отчеты и динамические графики (рис. 4), позволяющие оперативно анализировать изменения в работе системы. Такой подход обеспечивает прозрачность процессов и своевременное обнаружение возможных отклонений разработчиками, администраторами и преподавателями в ПИОС, что, в свою очередь, способствует поддержанию высокой стабильности и эффективности работы системы.

График общего потребления оперативной памяти, представленный на рисунке 4, дополнительно иллюстрирует устойчивость системы к высоким нагрузкам. Согласно экспериментальным данным, суммарное использование памяти моделей потокового машинного обучения не превышает 90 мегабайт, что является достаточно низким показателем. При этом наблюдается снижение потребления оперативной памяти при использовании комбинированной модели машинного обучения, что свидетельствует о ее эффективности и оптимизации ресурсов. Такая экономия памяти особенно актуальна при развертывании системы в условиях ограниченных вычислительных ресурсов, она обеспечивает стабильную работу ПИОС в реальном времени.

Заключение

В ходе исследования предложено архитектурное решение для внедрения моделей потокового машинного обучения на основе методов и алгоритмов потоковой обработки многомерных данных и машинного обучения в потоковой интеллектуальной образовательной системе. Экспериментально подтверждена высокая работоспособность разных моделей потокового машинного обучения в рамках предложенного архитектурного решения. Выведены метрики среднеквадратичной ошибки, коэффициента детерминации, а также общего использования памяти моделями.

Разработанное архитектурное решение позволило достичь поставленной цели, обеспечив платформу для персонализированного обучения и автоматической адаптации учебных материалов в реальном времени в ПИОС. В отличие от электронных образовательных систем, предлагаемая авторами ПИОС не ограничивается пакетной обработкой данных, а способна анализировать потоки данных (показатели поведения обучающихся) непрерывно, обеспечивая мгновенную корректировку траекторий обучения в помощь преподавателю.

Полученные экспериментальные результаты демонстрируют эффективность предложенной архитектуры потокового машинного обучения и открывают широкие перспективы для ее дальнейшего развития и практического применения. В рамках будущих исследований планируется пилотное внедрение разработанной системы в российских образовательных организациях с целью сбора и анализа данных обучающихся. Это позволит оценить работу модели в естественных условиях, проверить ее адаптивность к различным учебным сценариям и провести калибровку параметров для повышения точности предсказаний.

Список литературы

1. Евдокимова В.Е., Кириллова О.А. Возможности образовательных платформ для организации учебного процесса // *Современные наукоемкие технологии*. 2022. № 9. С. 120–125. DOI: 10.17513/snt.39319.
2. Ермаков С.Р., Мордвинов В.А. Применение функциональной синергетики для модернизации систем электронного обучения в учреждениях высшего образования // *Научно-технический вестник Поволжья*. 2023. № 7. С. 227–232. URL: <https://ntvprt.ru/> (дата обращения: 25.11.2024).
3. Montiel J., Halford M., Mastelini S.M., Bolmier G., Sourty R., Vaysse R., Zouitine A., Gomes H.M., Read J., Abdesaleem T., Bifet A. River: machine learning for streaming data in Python // *Journal of Machine Learning Research*. 2021. Vol. 22. Is. 110. P. 1–8. DOI: 10.48550/arXiv.2012.04740.
4. Aung T., Zaw H.T., Maw A.H., Mon M.T. Comprehensive Analysis: Monitoring Apache Kafka with Grafana, JMX Exporter, and Prometheus // *Proceedings of the 2024 5th International Conference on Advanced Information Technologies (ICAIT)*. 2024. Yangon, Myanmar: IEEE. 2024. P. 1–6. DOI: 10.1109/ICAIT65209.2024.10754944.
5. Junyi Academy Online Learning Activity Dataset. [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/junyiacademy/learning-activity-public-dataset-by-junyi-academy> (дата обращения: 25.11.2024).
6. Makowski D., Pham T., Lau Z.J., Brammer J.C., Lespinasse F., Pham H., Schölzel C., Chen S.H. NeuroKit2: A Python toolbox for neurophysiological signal processing // *Behavioral Research Methods*. 2021. Vol. 53. Is. 4. P. 1689–1696. DOI: 10.3758/s13428-020-01516-y.
7. Mishra L., Gupta T., Shree A. Online teaching-learning in higher education during lockdown period of COVID-19 pandemic // *International Journal of Educational Research Open*. 2020. Vol. 1. Article 100012. DOI: 10.1016/j.ijedro.2020.100012.
8. Romero C., Ventura S. Educational data mining and learning analytics: An updated survey // *WIREs Data Mining and Knowledge Discovery*. 2020. Vol. 10. No. 3. Article e1355. DOI: 10.1002/widm.1355.
9. Jónsson T., Pinson P., Nielsen H., Madsen H. Exponential Smoothing Approaches for Prediction in Real-Time Electricity Markets // *Energies*. 2014. Vol. 7. P. 3710–3732. DOI: 10.3390/en7063710.
10. Moharram H., Awad A., El-Kafrawy P.M. Optimizing ADWIN for steady streams // *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, Virtual Event: ACM*. 2022. P. 450–459. DOI: 10.1145/3477314.3507074.
11. Ермаков С.Р. Метод комбинированного обучения для потоковой интеллектуальной образовательной системы // *Научно-технический вестник Поволжья*. 2025. № 1. С. 180–185. URL: <https://ntvprt.ru/> (дата обращения: 11.11.2024).
12. Ермаков С.Р. Алгоритм инкрементального дерева решений для потоковой интеллектуальной образовательной системы // *Научно-технический вестник Поволжья*. 2024. № 12. С. 292–296. URL: <https://ntvprt.ru/> (дата обращения: 14.01.2025).
13. Watson G.L., Reid C.E., Jerrett M., Telesca D. Prediction & Model Evaluation for Space-Time Data [Электронный ресурс]. URL: <https://arxiv.org/abs/2012.13867> (дата обращения: 25.11.2024).