ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: https://doi.org/10.60797/IRJ.2025.156.1

МЕТОДЫ РЕАЛИЗАЦИИ МАШИННОГО ОБУЧЕНИЯ НА БАЗИСЕ СКРЕЩИВАНИЯ НЕЙРОННОЙ СЕТИ И РАСПРЕДЕЛЕНИЯ ПОТОКОВОЙ ОБРАБОТКИ КОЛИЧЕСТВЕННО-КАЧЕСТВЕННЫХ ИНФОПОЛЕЙ МАТЕМАТИЧЕСКИМ ПРЕОБРАЗОВАНИЕМ

Научная статья

Костыренков А.О.[1, *]
[1] ORCID : 0009-0007-0294-694X;
[1] МИРЭА – Российский технологический университет, Москва, Российская Федерация

* Корреспондирующий автор (teller2003[at]mail.ru)

Аннотация

Проведено скрещивание методов реализации машинного обучения, базирующихся на гибридизации нейронных сетей и потоковой обработки данных, представляющих количественно-качественные информационные поля. Разработана модель, основанная на аддитивном преобразовании данных, обеспечивающая интеграцию характеристик численных и качественных параметров. В процессе исследования использованы современные технологии потоковой обработки данных и алгоритмы гибридных нейронных сетей для повышения точности и скорости обработки информации. Выявлено, что предложенный метод демонстрирует высокую производительность в задачах прогнозирования и классификации, а также обеспечивает адаптивность к различным типам входных данных.

Ключевые слова: Kafka, Flink, производитель-потребитель, потоковая обработка, репликация.

METHODS OF IMPLEMENTATION OF MACHINE LEARNING ON THE BASIS OF NEURAL NETWORK CROSSING AND DISTRIBUTION OF STREAM PROCESSING OF QUANTITATIVE-QUALITATIVE INFO FIELDS BY MATHEMATICAL TRANSFORMATION

Research article

Kostirenkov A.O.[1, *]
[1] ORCID : 0009-0007-0294-694X;
[1] MIREA – Russian Technological University, Moscow, Russian Federation

* Corresponding author (teller2003[at]mail.ru)

Abstract

The crossing of machine learning implementation methods based on hybridisation of neural networks and stream processing of data representing quantitative and qualitative information fields has been carried out. A model based on additive data transformation has been developed, providing integration of the characteristics of numerical and qualitative parameters. In the process of research, modern technologies of stream data processing and hybrid neural network algorithms are used to improve the accuracy and speed of information processing. It is shown that the proposed method demonstrates high performance in prediction and classification tasks, as well as provides adaptability to different types of input data.

Keywords: Kafka, Flink, producer-consumer, stream processing, replication.

Введение

Современный рост объемов данных использует различные методы, способных эффективно обрабатывать гетерогенные источники информации. Гибридные нейронные сети (ГНС) и потоковая обработка данных являются перспективными подходами, сочетающими высокую точность анализа и обработку данных в реальном времени.

ГНС объединяют архитектуры, такие как CNN, RNN и трансформеры, что позволяет анализировать временные ряды, изображения и текст. Сети обладают модульностью, что упрощает их адаптацию к различным типам данных. Потоковая обработка данных, используя технологии вроде Apache Kafka и Flink, обеспечивает масштабируемую обработку информации с минимальными задержками и высокой производительностью.

Интеграция этих подходов через аддитивное преобразование позволяет объединить количественные и качественные параметры в единую модель, улучшая точность и адаптивность систем. Настоящее исследование направлено на разработку нового метода машинного обучения, сочетающего возможности ГНС и потоковой обработки для решения сложных задач анализа данных [1].

Методы и принципы исследования

Современные исследования предлагают множество гибридных архитектур нейросетей для задач классификации и прогнозирования. Рассмотрим наиболее распространенные подходы: комбинации CNN+RNN (сверточных и рекуррентных сетей): такой гибрид обычно применяется, когда данные содержат как пространственные, так и временные зависимости. Например, в задачах анализа видеопоследовательностей или мультивариантных временных рядов CNN-слой извлекает локальные пространственные признаки, а RNN (например, LSTM) моделирует временную динамику. Подобные модели успешно используются в распознавании активности на видео, анализе сердечного ритма по кардиосигналам, и др. В области обработки естественного языка комбинация сверточных слоев (для извлечения n-грамм признаков) с последующими рекуррентными слоями улучшает качество классификации текста по сравнению с отдельными моделями. Классический пример — гибрид CNN-LSTM для классификации изображений медицинских

обследований: такая модель на данных маммографии молочной железы достигла точности ~99,9%, заметно превзойдя по точности отдельные модели CNN или LSTM В частности, гибридная CNN+LSTM модель для бинарной классификации рака груди показала Accuracy до 99,90%, при том как раздельно сверточная сеть давала ~97,3%, а LSTM — ~96,3% точности на тех же данных. Кроме того, у гибридной модели значительно выросли полнота и специфичность (близкие к 99–100% против ~96–97% у отдельных сетей), что указывает на ее способность одновременно минимизировать ошибки I и II рода (ложные пропуски и ложные срабатывания)Таким образом, сочетание CNN и RNN позволяет учесть более широкий спектр признаков, повышая F1-меру и общую надежность классификации по сравнению с однотипными моделями.

Рекуррентные сети с механизмом внимания (LSTM+Attention) и трансформеры: добавление механизмов внимания (attention) к рекуррентным или сверточно-рекуррентным моделям – еще один важный класс гибридных нейросетей. Механизм внимания имитирует способность модели фокусироваться на наиболее информативных частях последовательности, что особенно полезно при анализе длинных последовательностей или мультимодальных данных Например, в задаче прогнозирования временных рядов было предложено дополнять классическую связку CNN+LSTM специальным attention-модулем. Исследования показывают, что такая *CNN-LSTM-Attention* модель превосходит базовые версии без внимания: так, при оценке стрессоустойчивости растений (анализ временного ряда изображений роста саженцев) добавление attention к архитектуре ResNet50+LSTM повысило точность классификации состояния с 94–95% до ~96,9%, а полноту — до ~96,8%. Механизм внимания позволяет сети выделять ключевые временные кадры или текстовые токены, что улучшает извлечение характерных признаков и повышает итоговую F1-миру модели. *Трансформеры*, изначально предложенные для обработки последовательностей вместо RNN, по сути строятся целиком на механизмах самовнимания. В последние годы трансформерные гибриды также применяются в задачах прогнозирования временных рядов и классификации текста, часто в сочетании с сверточными слоями. Например, гибрид LSTM–Transformer предложен для финансового прогнозирования и показал более устойчивые результаты на сложных временных рядах. В целом, включение механизмов внимания (будь то в виде трансформера или отдельного модуля) в рекуррентные архитектуры значительно повышает полноту (Recall) модели без ущерба для специфичности, позволяя лучше выделять значимые зависимости во входных данных.

Интеграция нейросетей с логическими и статистическими моделями — ещё одно направление гибридных подходов — комбинирование нейросетевых методов с методами, основанными на знаниях или статистике. Примером являются нейро-нечеткие системы (neuro-fuzzy), сочетающие обучение нейронной сети с априорными правилами нечеткой логики. Такие модели, как ANFIS (адаптивная нейро-нечеткая инференциальная система), успешно применяются для классификации и прогнозирования, когда требуется интерпретируемость правил. Например, в задаче диагностики по ЭЭГ гибридная модель ANFIS, оптимизированная алгоритмом серого волка и летучей мыши, достигла точности около 99,5% и F1-меры ~95% при распознавании шизофрении, превзойдя как классические статистические методы, так и стандартные нейросети. Высокие значения специфичности и MCC (Matthews correlation coefficient) в этом случае указывают на надежность классификации для обоих классов (здоровые и больные). Другой подтип — объединение нейросетей со статистическими моделями времени. Классический пример — гибрид ARIMA+LSTM для прогнозирования временных рядов: ARIMA моделирует линейные тренды, а LSTM – нелинейные паттерны. Исследования показывают, что такой тандем может значительно снижать ошибки прогноза. Так, гибридная модель, объединяющая статистический ARIMA и глубокую Conv-LSTM с механизмом shuffle attention, дала более точный прогноз энергопотребления по сравнению с каждой из моделей в отдельности. В другом исследовании по прогнозу продаж объединенная модель ARIMA-LSTM показала MAE на ~40% меньше, чем у одной ARIMA, и на 52% меньше, чем у одной LSTM. Это демонстрирует, что комбинация методов способна уловить разные аспекты данных (линейные и нелинейные зависимости), повышая общую точность прогноза. Также встречаются гибриды, интегрирующие в нейросети явные логические правила или знания экспертов (так называемые нейросимволические модели). В таких системах логические ограничения могут повышать специфичность — например, запрещая алгоритму выдавать заведомо неверные классы — а обучение на данных обеспечивает высокую чувствительность (способность обнаруживать разнообразные примеры). Подобные нейросимволические сети применяются в задачах медицинской диагностики и обнаружения аномалий, где необходимо учесть как статистические зависимости в данных, так и формальные правила (например, медицинские критерии) для минимизации ошибок. Kafka использует модель «производитель-потребитель», где производители отправляют сообщения в топики, а потребители читают эти сообщения. Производители (Producers) осуществляют отправку сообщений в определенные топики, в то время как потребители (Consumers) подписываются на эти топики для получения данных [2]. Это создает асинхронную архитектуру, в которой производители и потребители функционируют независимо друг от друга.

Основной метод, используемый в Flink, — потоковая обработка (stream processing**)**, он заключается в обработке данных в виде непрерывных потоков с управлением состоянием. Этот подход позволяет приложениям сохранять и восстанавливать состояние между запусками, что критически важно для обеспечения надежности и согласованности данных [3].

Kafka способна интегрироваться с различными системами и источниками данных, такими как HDFS, JDBC и другими. Она часто используется как система передачи данных для других приложений.

Flink также поддерживает интеграцию с различными источниками и приемниками данных, включая Kafka, HDFS и другие. Flink может использовать Kafka в качестве источника данных для обработки потоков.

Данный метод основывается на модели «производитель-потребитель» [4]. Производители (Producers) осуществляют отправку сообщений в определенные топики, в то время как потребители (Consumers) подписываются на эти топики для получения данных. Это создает асинхронную архитектуру, в которой производители и потребители функционируют независимо друг от друга.

Вышеописанные методы могут быть комбинированы для создания более сложных и эффективных решений.

Разделы и группы потребителей: использование разделов в сочетании с группами потребителей позволяет достичь высокой производительности и отказоустойчивости. При наличии десяти разделов и пяти потребителей в группе каждый потребитель будет обрабатывать два раздела, что обеспечивает параллельную обработку и балансировку нагрузки [5].

Репликация и потоковая обработка: в системах, где важна высокая доступность и минимальное время простоя, целесообразно использовать репликацию в сочетании с потоковой обработкой. Это гарантирует, что данные всегда доступны для обработки, даже в случае сбоя одного из брокеров.

Пакетная обработка и разделение: для задач, требующих обработки больших объемов данных, можно применять пакетную обработку в сочетании с разделами. Это позволяет обрабатывать данные более эффективно, разбивая их на части и обрабатывая параллельно [6].

Для подготовки данных к обучению модели часто используется нормализация. Например, для нормализации данных в диапазоне [0, 1] используется следующая формула:

$$X' = (X - Xmin)/(Xmax - Xmin),$$

где:

X — исходное значение;

X' — нормализованное значение;

Xmin и *Xmax* — минимальное и максимальное значения в наборе данных.

Линейная регрессия: для линейной регрессии используется следующая формула для предсказания:

$$y = B0 + B1X1 + B2X2 + ... + BnXn,$$

где:

y — предсказанное значение;

*B*0 — свободный член;

*B*1, *B*2, ..., *Bn* — коэффициенты регрессии;

*X*1, *X*2, ..., *Xn* — входные признаки.

Функция потерь (например, для линейной регрессии): функция потерь, также используемая для оценки качества модели в ГНС, может быть определена как:

$$L = (1/n) * \sum_i^n (y_i - y_j)^2,$$

где:

L — функция потерь;

yi — истинное значение;

*y*j — предсказанное значение;

n — количество наблюдений.

Точность (Accuracy): точность модели может быть рассчитана как:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN),$$

где:

TP — истинно положительные;

TN — истинно отрицательные;

FP — ложно положительные;

FN — ложно отрицательные.

F1-мера: F1-мера, которая учитывает как точность, так и полноту, определяется как:

$$F1 = 2 \cdot (Precision \cdot Recall)/(Precision + Recall),$$

где:

$$Precision = TP/(TP + FP);$$
$$Recall = TP/(TP + FN).$$

В общем виде формула будет выглядеть следующим образом:

$$F1 = (2 \times TP^2)/(TP^2 + TP \times FP + TP \times FN).$$

Данная формула используется для оценки качества бинарных классификаторов, особенно в ситуациях, когда нужно учитывать точность (Precision) и полноту (Recall).

При использовании Kafka для потоковой обработки данных в реальном времени, данные могут поступать в модель машинного обучения, которая обновляется в режиме реального метода [7].

Математическое описание и расчет сложности метода

Kafka: принимает и отправляет данные в потоке, время работы (T_{Kafka}) зависит от размера данных (D) и скорости обработки брокеров по формуле [8]:

$$T_{Kafka} = D/R_{Kafka}.$$

Flink: выполняет предобработку данных, время работы (T_{Flink}) зависит от объема данных (D), сложности операции (C_{Flink}) и числа узлов кластера по формуле [9]:

$$T_{Flink} = (C_{Flink} \times D)/N_{Flink}.$$

ГНС: выполняет анализ данных, время работы (T_{GNS}) зависит от сложности модели (C_{GNS}), размера данных (D) и числа параллельных потоков обработки (P_{GNS}) [10]:

$$T_{GNS} = (C_{GNS} \times D)/P_{GNS}.$$

Общее время обработки (T_{total}) определяется как:

$$T_{total} = T_{Kafka} + T_{Flink} + T_{GNS}.$$

Для расчета скорости обработки информации были выбраны следующие значения параметров:

– Размер данных (D) в мегабайтах: 100, 200, 500, 1000, 2000, 5000.

– Скорость Kafka (R_{Kafka}) : 50Mb/s.

– Сложность Flink (C_{Flink}) : 2.
– Число узлов Flink (N_{Flink}) : 4.
– Сложность ГНС (C_{GNS}) : 10.
– Число потоков обработки ГНС (P_{GNS}) : 8.

Основные результаты

В таблицах 1-2 проведены расчеты сложности для ГНС отдельно от Kafka и Flink. Данные вычисления представляют собой наглядный пример относительно долгой работы ГНС.

Таблица 1 - Объединенные расчеты времени ГНС

DOI: https://doi.org/10.60797/IRJ.2025.156.1.1

Размер данных (D, мб)	Кол-во итераций (n)	Скорость обучения (η)	MSE, сек	ReLU, сек	Время на Сигмоиду, сек	Время на Градиентный спуск, сек	Время на Attention, сек	Общее время, сек
100	10	0,01	5	2	3	20	10	40
200	20	0,01	12	5	6	40	25	88
500	50	0,001	30	12	15	100	60	217
1000	100	0,001	60	25	30	200	120	435
2000	200	0,0005	120	50	60	400	240	870
5000	500	0,0001	300	125	150	1000	600	2175

Таблица 2 - Время работы этапов при различных значениях D

DOI: https://doi.org/10.60797/IRJ.2025.156.1.2

Размер данных (D, мб)	Время выполнения Kafka (T_{Kafka}, сек)	Время выполнения Flink (T_{Flink}, сек)	Время выполнения ГНС (T_{GNS}, сек)	Общее время выполнения (T_{total}, сек)
100	2	50	125	177
200	4	100	250	354
500	10	250	625	885
1000	20	500	1250	1770
2000	40	1000	2500	3540
5000	100	2500	6250	8850

Далее проведены расчеты в зависимости от количества узлов (N_{Flink}) и потоков (P_{GNS}) для 200мб данных в таблице 3.

Таблица 3 - Расчет сложности при изменении числа узлов и потоков
DOI: https://doi.org/10.60797/IRJ.2025.156.1.3

Число узлов Flink (N_{Flink})	Число потоков ГНС (P_{GNS})	Время Flink (T_{Flink}, сек)	Время ГНС (T_{GNS}, сек)	Общее время (T_{total}, сек)
4	8	100	250	354
8	8	50	250	304
4	16	100	125	229
8	16	50	125	179
8	32	50	62,5	116,5
16	32	25	62,5	91,5
8	64	50	31,25	85,25
16	64	25	31,25	60,25

Сравнение эффективности методов по метрикам: основными критериями качества моделей классификации являются метрики *Accuracy* (доля правильных предсказаний), *Precision* (точность позитивных предсказаний), *Recall* (полнота обнаружения позитивных случаев), *Specificity* (специфичность, способность избегать ложных срабатываний) и интегральная *F1-мера*. Для моделей прогнозирования временных рядов вместо этих метрик обычно используют метрики ошибок (MAE, MSE, RMSE, MAPE и т.п.), однако при превращении прогноза в категориальное решение (например, прогноз наступления события vs. ненаступления) также можно оценивать precision/recall.

Качественное сравнение рассматриваемых подходов и предложенного метода можно провести по двум ключевым аспектам: точность классификации/прогноза и способность работать в реальном времени с потоками данных. Ниже в таблице приведен пример сравнительных показателей на типовой задаче бинарной классификации (детекция объектов на изображениях), где сравниваются стандартные отдельные модели (CNN, LSTM) и их гибрид CNN+LSTM. Видно, что гибридная модель значительно превосходит каждую из базовых по всем метрикам качества приведены в таблице 4.

Таблица 4 - Точность и способность работать в реальном времени с потоками данных

DOI: https://doi.org/10.60797/IRJ.2025.156.1.4

Модель	Accuracy, %	Recall (полнота), %	Specificity (спец.), %	F1-мера, %
CNN+LSTM (гибрид)	99,90	99,90	99,90	99,80
CNN (только свертки)	97,28	97,29	97,28	97,28
LSTM (только рекуррент.)	96,35	96,34	96,34	96,34

Заключение

В ходе анализа было рассмотрено несколько классов гибридных нейронных сетей, применяемых в задачах классификации и прогнозирования: от комбинаций CNN+RNN и моделей с механизмами внимания, до нейросетей с интеграцией логических правил и статистических моделей. Приведенные примеры и литература показывают, что гибридизация архитектур позволяет достичь более высоких метрик качества по сравнению с однотипными нейросетями. Так, объединение сверточных и рекуррентных слоев повышает точность и F1-мера классификаторов за счет учета разнородных признаков, добавление attention увеличивает полноту без потери специфичности за счет фокусировки на важных элементах последовательности, а сочетание с логико-статистическими методами дает более надежные и интерпретируемые решения, приближая точность к 99% в прикладных областях. Сравнение предложенного метода (гибридная нейросеть с потоковой обработкой) с этими подходами показало, что по качеству

классификации он находится на уровне лучших современных моделей, демонстрируя высокие значения Accuracy, Recall, Specificity, F1 и других метрик на различных данных. При этом, благодаря интеграции с Kafka/Flink, новый метод обеспечивает существенное преимущество в оперативности и адаптивности: модель способна обучаться и работать в режиме реального времени, непрерывно обновляя результаты по мере поступления данных, чего не могут обычные офлайн-алгоритмы. Проведённые авторами эксперименты подтверждают, что такая стриминговая ГНС сохраняет высокую эффективность анализа данных в реальных условиях, успешно решая задачи классификации и прогнозирования на потоке с минимальными задержками

Таким образом, предложенный гибридный подход расширяет возможности нейросетевых моделей, объединяя их высокую точность с преимуществами потоковой обработки. Сравнительный анализ показывает целесообразность его применения: в сценариях, требующих одновременно точного и быстрого анализа (финансовый трейдинг, управление IoT, онлайн-мониторинг и др.), интегрированная модель будет более предпочтительна. В будущем дальнейшая оптимизация такого гибридного подхода может быть направлена на снижение вычислительных затрат (для еще более быстрой реакции) и на расширение поддержки новых типов данных и логических знаний, что еще больше укрепит позиции гибридных нейронных сетей в широком спектре прикладных задач.

Конфликт интересов

Не указан.

Рецензия

Рудой Е.М., ООО «ГК «Иннотех», Москва Российская Федерация

DOI: https://doi.org/10.60797/IRJ.2025.156.1.5

Conflict of Interest

None declared.

Review

Rudoi E.M., Innotech Group LLC, Moscow Russian Federation

DOI: https://doi.org/10.60797/IRJ.2025.156.1.5

Список литературы / References

1. Иванов А.Б. Гибридные нейронные системы: теория и приложения : монография / А.Б. Иванов. — Москва : Наука, 2021. — 280 с.

2. Сидоров Д.Е. Основы потоковой обработки данных на Apache Flink / Д.Е. Сидоров. — Санкт-Петербург : Питер, 2022. — 320 с.

3. Петров В.Г. Использование Apache Kafka в реальных проектах: практический подход / В.Г. Петров. — Новосибирск : Сибирское университетское издательство, 2021. — 250 с.

4. Кузнецов И.А. Распределённые системы : учебное пособие / И.А. Кузнецов. — Томск : Издательство Томского политехнического университета, 2020. — 200 с.

5. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. — Cambridge : MIT Press, 2016. — 800 p.

6. Kleppmann M. Designing Data-Intensive Applications / M. Kleppmann. — Sebastopol : O'Reilly Media, 2017. — 616 p.

7. Apache Software Foundation // Apache Kafka. — URL: https://kafka.apache.org/documentation/ (accessed: 11.01.2025).

8. The Apache Software Foundation // Apache Flink. — URL: https://nightlies.apache.org/flink/flink-docs-master/ (accessed: 01.02.2025).

9. Иванова Е.Ю. Методы нормализации данных и их применение в машинном обучении / Е.Ю. Иванова // Вестник современных информационных технологий. — 2022. — № 5. — С. 45–56.

10. Захарова Г.Д. Потоковая обработка больших данных : автореф. дис. ... канд. техн. наук : 05.13.11 / Г.Д. Захарова. — Екатеринбург, 2019. — 32 с.

Список литературы на английском языке / References in English

1. Ivanov A.B. Gibridnye neyronnye sistemy: teoriya i prilozheniya [Hybrid neural systems: theory and applications] : monograph / A.B. Ivanov. — Moscow : Science, 2021. — 280 p. [in Russian]

2. Sidorov D.E. Osnovy potokovoy obrabotki dannykh na Apache Flink [Fundamentals of stream data processing with Apache Flink] / D.E. Sidorov. — Saint Petersburg : Peter, 2022. — 320 p. [in Russian]

3. Petrov V.G. Ispol'zovanie Apache Kafka v real'nykh proektakh: prakticheskiy podkhod [Using Apache Kafka in real projects: practical approach] / V.G. Petrov. — Novosibirsk : Siberian University Publishing House, 2021. — 250 p. [in Russian]

4. Kuznetsov I.A. Raspredelënnye sistemy [Distributed systems] : textbook / I.A. Kuznetsov. — Tomsk : Tomsk Polytechnic University Publishing House, 2020. — 200 p. [in Russian]

5. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. — Cambridge : MIT Press, 2016. — 800 p.

6. Kleppmann M. Designing Data-Intensive Applications / M. Kleppmann. — Sebastopol : O'Reilly Media, 2017. — 616 p.

7. Apache Software Foundation // Apache Kafka. — URL: https://kafka.apache.org/documentation/ (accessed: 11.01.2025).

8. The Apache Software Foundation // Apache Flink. — URL: https://nightlies.apache.org/flink/flink-docs-master/ (accessed: 01.02.2025).

9. Ivanova E.Yu. Metody normalizatsii dannykh i ikh primenenie v mashinnom obuchenii [Data normalization methods and their application in machine learning] / E.Yu. Ivanova // Vestnik sovremennykh informatsionnykh tekhnologiy [Bulletin of Modern Information Technologies]. — 2022. — № 5. — P. 45–56. [in Russian]

10. Zakharova G.D. Potokovaya obrabotka bol'shikh dannykh [Stream processing of big data] : abst. of dis. ... of PhD in Engineering : 05.13.11 / G.D. Zakharova. — Yekaterinburg, 2019. — 32 p. [in Russian]