

Сравнительный анализ рекуррентных сетей и трансформерных моделей в предиктивном мониторинге

А.Н. Кошелев, К.Э. Привалов, С.А. Петров, В.К. Денисенко,

Национальный исследовательский университет «МЭИ»

Аннотация: Работа сравнивает применение рекуррентных сетей и моделей на основе архитектуры трансформеров для решения задачи предсказания времени завершения бизнес-процесса. Модели по определению моделируют последовательность исполнения действий и способны учитывать множество атрибутов в определении целевых характеристик. Для сравнения были использованы рекуррентная модель долгой краткосрочной памяти и трансформер-энкодер собственной архитектуры, работа которых была опробована на открыто представленных реальных данных из логов службы поддержки. Обучение и тестирование моделей проводилось с помощью языка Python с помощью библиотек pandas, numpy, torch с одинаковой для обеих моделей подготовкой данных, формированием префиксов и разбиением по времени. Сравнение в результате экспериментов по средней абсолютной ошибке показало преимущество энкодера-трансформера; по среднеквадратичному отклонению зафиксирована примерно одинаковая точность моделей с незначительно большей точностью модели-трансформера.

Ключевые слова: предиктивный мониторинг, журнал событий, машинное обучение, трансформер-энкодер, нейронные сети, подготовка данных, регрессионная модель, нормализация, паддинг, рекуррентная сеть, архитектура модели.

Введение

Соблюдение сроков исполнения задачи (также называемой кейсом), установленных договором о стандартах качества услуг, является наиболее экономически важной задачей компаний. Анализ и оптимизация внутренних бизнес-процессов позволяет компаниям укладываться в обозначенные сроки и выполнять больше услуг за меньшее время, из-за чего компания не несет потери в виде неустоек и увеличивает свою прибыль ввиду увеличения эффективности [1].

Таким образом, предсказание времени, необходимого для завершения определенной задачи, и причин его увеличения становится основной задачей подобного анализа. Решение этой задачи исторически сводилось к ручному изучению процессов и методу экспертных оценок, которые лично проводились соответствующими работниками. Такие методы оказались

субъективными, затратными по времени и труду и не могли быть масштабированы.

Им на смену были выдвинуты базовые статистические методы (например, вычисление среднего времени переходов по состояниям), которые обеспечивали определение оставшегося времени, но не описывали факторов, влияющих на продолжительность, не учитывали контекст и вариативность в исполнении процесса [2]. Классические методы машинного обучения, такие как линейная регрессия, деревья решений, случайный лес, модели выживаемости (градиентный бустинг и модель пропорциональных рисков Кокса) показали приемлемую точность только при проведенной подготовке данных, включающей ручную подготовку признаков (feature engineering), определение правильного подхода к кодированию признаков и подбор гиперпараметров, что увеличивает трудозатраты при подготовке модели [3].

Так как даже с доработкой классические методы машинного обучения упрощают представление последовательности событий и теряют точность при увеличении длины трассы и насыщенности контекста, справедливо для решения задачи прогноза оставшегося времени завершения кейса применять последовательные нейросетевые модели: модель длинной краткосрочной памяти (long short-term memory – LSTM) и компактную модель на основе архитектуры трансформера, которые напрямую учитывают последовательности событий и атрибуты кейса.

Будучи рекуррентной моделью, LSTM пошагово считывает историю событий и просчитывает зависимости внутри контекста даже в длинных трассах [4]. В силу того, что применение этой модели в рамках задачи предиктивного мониторинга уже распространено, ее показатели по точности, выявленные в ходе экспериментов, будут выступать референсными значениями для определения точности более нового подхода: трансформера-энкодера [5]. Модели, построенные на архитектуре трансформера,

используют механизм внимания, который в контексте этой задачи позволяет модели наблюдать за последовательностью событий целиком и выделять связи в событиях на большем расстоянии между событиями. Кроме того, существующие исследования применения моделей в других областях показывают, что подход на основе трансформеров может выделять более нелинейные паттерны, которые могут быть упущены рекуррентными моделями [6].

Таким образом, целью работы является сопоставление по точности предсказания этих двух моделей. Достижение этой цели требует выполнения следующих задач: выбор методики подготовки данных и обучения моделей, выбор архитектуры и разработка модели-трансформера и LSTM-модели, выбор метрик для сравнения по точности предсказания, проведение сравнения.

Общая подготовка данных

В целях точности сравнения моделей данные, которые будут использоваться для обучения и тестирования моделей, должны быть подготовлены одинаково.

Вне зависимости от конкретного набора данных для обучения и тестирования требуется последовательно выполнить ряд операций для адаптации данных перед их применением на последовательных моделях: фильтрация кейсов с количеством событий меньше двух, приведение категориальных значений к числовым, сортировка данных по временной отметке, регуляция длины кейсов [7]. Кроме того, необходимо внести и рассчитать временные признаки, служащие ключевыми для обучения моделей: длительность с начала кейса для каждого входящего в него события, время между событиями. Для числовых признаков важно выполнить нормализацию с целью уменьшения влияния значений с большим, чем остальные, порядком или значением. В качестве метода нормализации

применен метод «RobustScaler», широко используемый для масштабирования признаков с присутствием выбросов [8].

В качестве целевой переменной выбрано оставшееся время до конца исполнения кейса, которое в общем виде рассчитывается в формуле (1).

$$y_{c,k} = t_{c,T_c} - t_{c,k}, \quad (1)$$

где c – индекс кейса, T_c – число событий кейса, k – длина префикса, $t_{c,i}$ – отметка времени i -го события кейса, t_{c,T_c} – время последнего события кейса, $t_{c,k}$ – время последнего события в префиксе длины k , $y_{c,k}$ – остаточное время до конца кейса.

Кроме того, чтобы модель понимала динамику выполнения кейса, необходимо как вход для модели также рассчитать время, прошедшее с момента предыдущего события, что должно выявить паузы, задержки и наиболее интенсивные моменты.

Последовательные модели для своего обучения требуют префиксы фиксированной длины, далее именуемой L_{max} . Значение этого параметра определяется как 95-й перцентиль длины кейсов. Так как в жизни количество этапов выполнения кейса может варьироваться, может возникнуть два случая с длиной кейса L_c , при которых требуется дополнительная обработка данных: $L_c < L_{max}$, $L_c > L_{max}$. Первый случай предполагает введение паддингов и масок для префиксов обучения: паддингом выполняется увеличение кейса до длины L_{max} путем добавления в конец последовательности специальных нулевых значений для признаков; маска же – это вектор индикаторов, отображающий, какие позиции соответствуют реальным событиям, а какие являются паддингом, и рассчитываемый для LSTM по формуле (2).

$$m_{c,j,k} = \begin{cases} 1, & \text{если } 1 \leq j \leq k, \\ 0, & \text{если } k \leq j \leq L_{max}, \\ & j = 1, \dots, L_{max} \end{cases} \quad (2)$$

где c – индекс кейса; $m_{c,j,k} \in \{0,1\}$ – значение маски: 1 означает реальное значение для позиции j в префиксе длины k , а 0 – паддинг. Для модели-трансформера применяется та же маска, но в логическом и инвертированном виде: значение «true» (оно же в числовом виде 1) используется для обозначения паддинга.

Второй же случай предполагает, что для обучения модели из событий будут составлены префиксы, не превышающие по длине L_{max} , то есть события, чей порядковый номер в кейсе больше L_{max} , не будут анализироваться моделью при обучении, иначе модель будет переобучена под длинные кейсы и, возможно, упустит релевантные последовательности и закономерности [9].

Таким образом, для обучения модель будет оперировать префиксами кейсов – последовательностями событий, каждое из которых имеет свои характеристики, отсортированных по времени и сгруппированных по идентификатору кейса. По каждому кейсу модель будет учиться на нескольких префиксах длиной от 2 до L_{max} . Описанная схема подготовки данных является общей для обеих рассматриваемых моделей, что обеспечит сопоставимость результатов их тестирования.

Выбор архитектуры и разработка модели-трансформера и LSTM-модели

В качестве архитектур последовательных моделей для задачи прогнозирования оставшегося времени были выбраны многослойная LSTM-сеть и трансформер-энкодер. LSTM-модели традиционно используются для подобного прогнозирования, трансформеры же предлагаются как более гибкая альтернатива для выявления дальних зависимостей [10].

LSTM-модель спроектирована по схеме «многие-к-одному», принятой в работах по предиктивному мониторингу [10]. Базовый уровень модели реализован модулем «nn.LSTM» библиотеки «PyTorch» и обеспечивает

вычисление непосредственно рекуррентных состояний и обработку многослойности. Для соответствия целям исследования модель была доработана слоем эмбедингов для работы с категориальными признаками; было разработано объединение признаков во входной вектор признаков события; добавлено 2 слоя LSTM с размерностью скрытого состояния 128; осуществлен выбор представления префикса и добавлен регрессионный выход для перевода выходного значения последнего слоя в оценку по времени.

Архитектура модели-трансформера основана на архитектуре энкодера «Attention is all you need» и его адаптациях для предиктивного мониторинга. В качестве базового блока используется стандартный «nn.TransformerEncoder» с несколькими слоями «TransformerEncoderLayer», реализованных в PyTorch. Этот базовый блок был доработан общим слоем эмбедингов и признаков; позиционным кодированием для различения порядковых позиций в префиксе; 2 слоями трансформера-энкодера с размерностью модели 128, четырьмя головами внимания и нейросетью с прямой связью (feedforward neural network) размерностью 256; агрегацией представления префикса на выходе энкодера; аналогичным LSTM регрессионным выходом на основе «ReLU».

Такие гиперпараметры как размерность эмбедингов, размерности состояний, количество слоев, число голов, скорость обучения подбирались вручную на основе перебора, причем суммарное количество параметров и глубина сети определялись так, что количество параметров оставалось одного порядка, чтобы различия в качестве предсказаний было нельзя объяснить размером модели. Использовался оптимизатор «AdamW» с фиксацией диапазона начальных скоростей обучения, размера пачек для обучения и с ранней остановкой по средней абсолютной ошибке (mean

absolute error – MAE) на валидационной выборке: обучение заканчивалось, если эта метрика начинала повышаться или выходила на плато.

Обучение моделей и проведение эксперимента

Датасет взят из открытых источников и представляет собой событийный лог по процессу управления тикетами службы поддержки компании, разрабатывающей программное обеспечение. Лог содержит сведения о 21 тыс. событий по 4580 задачам (кейсам) с 14 видами активностей событий. Среди характеристик по каждому событию присутствуют название рабочей группы, ответственной за выполнение заявки; заказчик; продукт; временная метка события (когда оно произошло); приоритет заявки и прочие.

Анализ датасета показал, что длительности выполнения кейсов варьируются от ~30,6 до ~60 дней с медианным значением длины кейса 39,85 дней. В связи с этим единицей измерения целевой переменной предсказания – времени окончания кейса – были выбраны именно дни.

Средняя длина кейса составила 4,66 события, а 95-й перцентиль длины кейсов – 7. Соответственно, значение L_{max} было установлено равному 7.

Данные были подготовлены по описанной ранее методике и разбиты на три непересекающихся выборки: тренировочную, валидационную и тестовую, содержащие соответственно 70%, 10% и 20% от общего числа кейсов. Таким образом, каждая выборка включала полные трассы кейсов, что исключило утечки информации между наборами.

В качестве метрик по оценке качества была выбрана MAE, упомянутая ранее, ввиду ее устойчивости к выбросам и простотой в интерпретации в качестве «средней ошибки в днях». Кроме того, в качестве метрики было так же выбрано среднеквадратичное отклонение (root mean square error – RMSE), позволяющее более точно учитывать крупные ошибки модели за счёт квадратичного роста вклада больших ошибок [11]. Средняя абсолютная

ошибка в процентах как метрика не была использована, поскольку при малых значениях истинного оставшегося времени она дает неадекватно завышенные значения ошибки и может вводить в заблуждение при интерпретации информации [11].

Обучение обеих моделей на одинаковых данных показало, что модель-трансформер точнее в своих предсказаниях на обучающей и валидационных выборках, чем LSTM-модель.

На рис. 1 приведен график зависимости MAE от эпохи обучения на обучающей и валидационной выборках, который показывает, что модель-трансформер использовала больше эпох для сходимости и смогла достичь меньшего значения MAE.

Преимущество модели-трансформера в данном эксперименте можно интерпретировать как результат более эффективного использования информации об атрибутах событий.

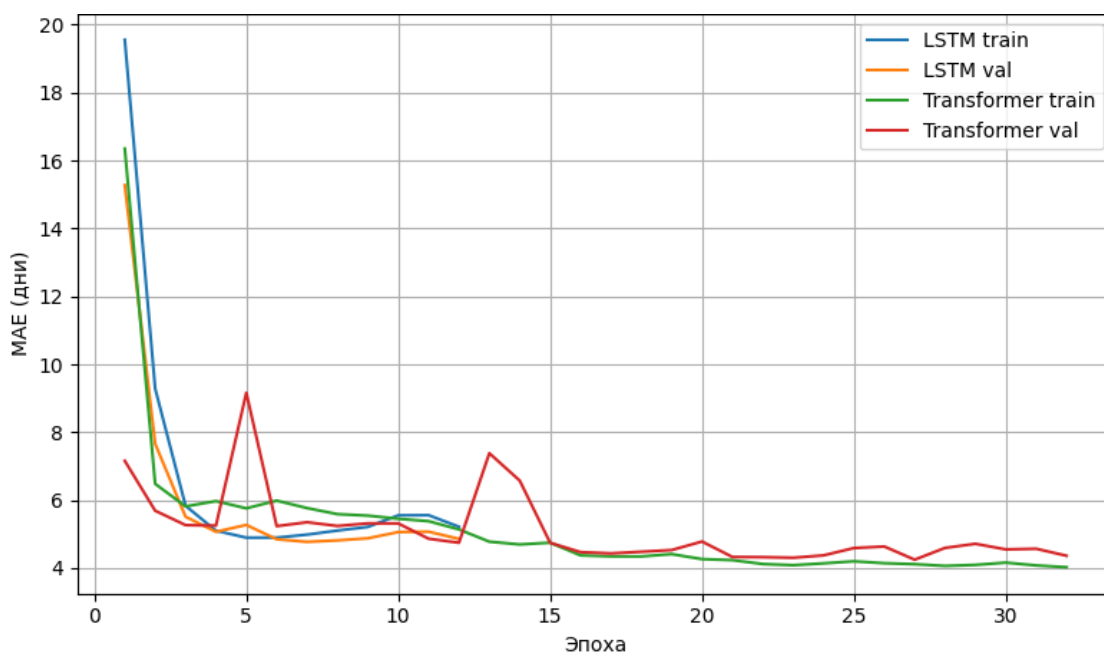


Рис. 1 – Динамика изменения метрики MAE при обучении моделей на обучающей и валидационной выборках

Сравнение точности предсказаний предполагаемого времени окончания кейса для обеих моделей на всех трех выборках по метрике MAE

приведено в таблице №1, а по метрике RMSE – в таблице 2. Значения метрик приведены в днях.

Таблица №1

Сравнение точности предсказаний моделей по метрике MAE

Наименование выборки	Наименование модели	
	LSTM	Трансформер
Тренировочная	4,497525	3,939955
Валидационная	4,859829	4,363508
Тестовая	5,327726	4,321012

Разница на тестовой выборке по метрике MAE составляет 1,01 дня в пользу модели-трансформера. Относительно медианной длины кейса ошибка модели-трансформера составляет 10,84%, а ошибка модели LSTM – 13,37%. Таким образом, трансформер показывает точность выше, чем LSTM, в абсолютном сравнении ошибок.

Таблица №2

Сравнение точности предсказаний моделей по метрике RMSE

Наименование выборки	Наименование модели	
	LSTM	Трансформер
Тренировочная	6,880469	6,348772
Валидационная	7,583694	7,046789
Тестовая	7,085173	6,540916

Разница на тестовой выборке по метрике RMSE между моделями составляет 0,55 дня – существенно меньше, чем по MAE, что указывает на сопоставимый уровень крупных ошибок у обеих моделей. При этом модель-трансформер и в этом сравнении показывает меньшую ошибку.

Заключение

В рамках работы была разработана единая методика подготовки данных, разработана собственная архитектура моделей LSTM и трансформера-энкодера, было проведено сравнение моделей по предсказанию оставшегося времени до завершения кейса при одинаковой подготовке данных. Результаты экспериментов показали, что на рассматриваемом датасете модель-трансформер обеспечивает более низкую метрику ошибки MAE, притом значение ошибки по RMSE отличаются незначительно и указывают на паритет моделей в ошибках с большим значением. Таким образом, на рассматриваемом датасете трансформер-энкодер демонстрирует более низкую MAE и RMSE, что свидетельствует о его лучшей способности использовать информацию длинных префиксов и богатого набора атрибутов.

Полученные результаты сравнения позволяют считать модель-трансформер применимой для задачи предиктивного мониторинга. Однако важно отметить, что подбор архитектур и гиперпараметров осуществлялся вручную для поддержания одинакового уровня количества параметров моделей, а обучение проходило по простому протоколу «AdamW». Методы более мощного поиска параметров (байесовский или рандомный поиск), а также проведение экспериментов на других наборах данных могут показать лучшее качество обеих моделей с отличающимся относительным разрывом по метрикам ошибок.

Литература

1. Гаглюева И.Э., Ковалева М.А., Саханский Ю.В., Джериев Ч.А. Разработка алгоритма оптимизации бизнес-процессов ИТ-компании и модели информационной системы поддержки принятия решений // Инженерный вестник Дона. 2023. №10. URL: ivdon.ru/magazine/archive/n10y2023/8761.

2. Käppel M., Jablonski S., Schöning S. Evaluating Predictive Business Process Monitoring Approaches on Small Event Logs. QUATIC. Cham. Springer. 2021. Pp. 167-182.

3. Heaton J. An empirical analysis of feature engineering for predictive modeling. SoutheastCon. Norfolk. IEEE. 2016. Pp. 1-6.

4. Вихтенко Э.М., Глеков М.С. Исследование моделей рекуррентных нейронных сетей для прогнозирования уровня реки на примере данных о реке Амур // Инженерный вестник Дона. 2025. №3 URL: ivdon.ru/magazine/archive/n3y2025/9913.

5. Navarin N., Vincenzi B., Polato M., Sperduti A. LSTM Networks for Data-Aware Remaining Time Prediction of Business Process Instances. IEEE Symposium Series on Computational Intelligence (SSCI). Honolulu. IEEE. 2017. Pp. 1-7.

6. Pölz, A., Blaschke A., Komma J., Farnleitner A., Derx J. Transformer Versus LSTM: A Comparison of Deep Learning Models for Karst Spring Discharge Forecasting. Water Resources Research. 2024. V. 4. №60. Pp. 10-29.

7. Сай Ван К., Щербаков М.В. Прогнозирование отказов сложных многообъектных систем на основе комбинации нейросетей: пути повышения точности прогнозирования. Прикаспийский журнал: управление и высокие технологии. 2020. Т. 49. №1. Сс. 49-60.

8. Клюковкин Г.К. Сравнение методов масштабирования признаков: StandardScaler, MinMax, Robust. Вестник науки и творчества. 2023. Т. 91. №9. Сс. 32-29.

9. Teinema I., Dumas M., La Rosa M., Maggi F. Outcome-Oriented Predictive Process Monitoring: Review and Benchmark. ACM Transactions on Knowledge Discovery from Data. 2017. V. 13. №2. Pp. 1-57.

10. Tax N., Verenich I., La Rosa M., Dumas, M. Predictive business process monitoring with LSTMs. Benelux Conference on Machine Learning. Eindhoven. Springer. New York. 2017. Pp. 477-492.
11. Bolt A., Sepúlveda M. Process Remaining Time Prediction using Query Catalogs. Business Process Management Workshops. Springer. Cham. 2013. Pp. 54-65.

References

1. Gagloeva I.E., Kovaleva M.A., Sakhanskiy Yu.V., Dzheriev Ch.A. Inzhenernyj vestnik Dona. 2023. №10. URL: ivdon.ru/magazine/archive/n10y2023/8761.
2. Käppel M., Jablonski S., Schöning S. Evaluating Predictive Business Process Monitoring Approaches on Small Event Logs. QUATIC. Cham. Springer. 2021. Pp. 167-182.
3. Heaton J. An empirical analysis of feature engineering for predictive modeling. SoutheastCon. Norfolk. IEEE. 2016. Pp. 1-6.
4. Vikhtenko E.M., Glekov M.S. Inzhenernyj vestnik Dona. 2025. №3. URL: ivdon.ru/magazine/archive/n3y2025/9913
5. Navarin N., Vincenzi B., Polato M., Sperduti A. LSTM Networks for Data-Aware Remaining Time Prediction of Business Process Instances. IEEE Symposium Series on Computational Intelligence (SSCI). Honolulu. IEEE. 2017. Pp. 1-7.
6. Pölz, A., Blaschke A., Komma J., Farnleitner A., Derx J. Transformer Versus LSTM: A Comparison of Deep Learning Models for Karst Spring Discharge Forecasting. Water Resources Research. 2024. V. 4. №60. Pp. 10-29.
7. Say Van K., Shcherbakov M.V. Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii. 2020. V. 49. №1. Pp. 49-60.
8. Klyukovkin G.K. Vestnik nauki i tvorchestva. 2023. V. 91. №9. Pp. 32-29.



9. Teinema I., Dumas M., La Rosa M., Maggi F. Outcome-Oriented Predictive Process Monitoring: Review and Benchmark. ACM Transactions on Knowledge Discovery from Data. 2017. V. 13. №2. Pp. 1-57.
10. Tax N., Verenich I., La Rosa M., Dumas, M. Predictive business process monitoring with LSTMs. Benelux Conference on Machine Learning. Eindhoven. Springer. New York. 2017. Pp. 477-492.
11. Bolt A., Sepúlveda M. Process Remaining Time Prediction using Query Catalogs. Business Process Management Workshops. Springer. Cham. 2013. Pp. 54-65.

Дата поступления: 6.12.2025

Дата публикации: 6.01.2026