Оглавление

Глава 1
Введение

Предсказание набора классов более применима на практике, в отличие от простой классификации, поскольку все, что окружает нас в реальном мире, имеет несколько меток [42]. Эта аналогия также может быть перенесена на последовательности категориальных значений. Каждый элемент последовательностей содержит вектор с несколькими метками с единицами в позициях, присутствующих для этого элемента, и нулями для тех, которые отсутствуют, и дополнительной информацией.

Однако задача классификации с несколькими метками имеет свои недостатки. Должен быть достаточно большой набор данных, чтобы алгоритм мог выучить различные варианты событий. Это требует значительных вычислительных ресурсов и больших моделей.

Существует множество подходов к решению задач классификации с несколькими метками. Они используются в различных областях: компьютерное зрение [16], обработка естественного языка [86] или классические алгоритмы для табличных данных [72].

Предсказание набора классов также используется для временных рядов в качестве входных данных для алгоритмов. В целом, связь между объектом в разных временных метках делает модель выразительной и мощной для решения задач с последовательными данными. Существует несколько подходов таких, как трансформер или рекуррентная нейронная сеть, для поиска зависимостей. Однако применение современных методов глубокого обучения ограничено [101] и фокусируется на метках прогнозирования для последовательности в целом, в некоторых случаях пытаясь ограничить объем требуемой информации [21]. Это понятно, потому что в целом это сложная задача, чтобы превзойти классическое машинное обучение в общем сценарии [75]. Мы приписываем этому соединению такое название, как граф последовательности.

Еще одна связь, которую стоит рассмотреть,- это связь между различными метками и необходимость учитывать корреляцию между ними [20], что не выполняется автоматически разными моделями. Такое соединение между сущностями называется графом меток.

В данной работе оценивается корреляция между метками в последовательной задаче мультиклассовой классификации: модель делает прогнозирование меток, которые встречаются на следующем шаге, принимая во внимание предыдущие метки. Как граф последовательности, так и граф меток важны для последовательного прогнозирования с несколькими метками.

Происходит объединение информации о предыдущем элементе в последовательности и передача ее в качестве входных данных в наш транс-

формер. Модель использует механизм собственного внимания для меток с целью получения прогноза. Мы включаем в нашу архитектуру другие элементы, которые часто встречаются при последовательной классификации с несколькими метками.

Перечислим основные аспекты, составляющие новизну предложенного подхода:

1. В данной работе предложена новая архитектура LANET, предназначенная для предсказания множества меток следующего события на основе анализа предшествующих событий. Ключевая инновация подхода заключается в особом способе обработки исторических данных перед их подачей в блок с механизмом self-attention.

2. Проведено детальное сравнительное исследование эффективности LANET и современных проверенных моделей прогнозирования временных последовательностей. Экспериментальные результаты демонстрируют существенное преимущество LANET над аналогами, достигнутое благодаря оптимизированной структуре входных данных.

3. В ходе анализа компонентов LANET выявлено, что наибольшее влияние на качество прогнозирования оказывает учет взаимосвязей между метками, тогда как временные характеристики событий играют второстепенную роль.

4. Доказана эффективность агрегирования по меткам над агрегированием по времени.

5. Представлены теоретические результаты применения данного агрегирования к процессам Хокса.

Работа разделена по главам следующим образом.

- В гл. 2 описывается постановка задачи, вводятся обозначения, которые будут использоваться в последующих главах.

- В гл. 3 проводится подробный обзор существующих решений, для каждого описываются проблемы которые решаются нашими методами.

- В гл. 4 приводится теоретическая база нашего подхода, в том числе краткое описание используемых существующих методов.

- В гл. 5 описывается техническая часть: параметры численных экспериментов, вычислительные мощности и используемые программные пакеты.

- В гл. 6 мы демонстрируем результаты работы и сравниваем наши методы с классическими подходами.
- Наконец, в гл. 7 подводятся итоги исследования, выделены положительные стороны и направления для дальнейшей работы.

Глава 2
Постановка задачи

В классической теории обработки последовательностей событий каждое событие традиционно характеризуется одной категориальной меткой и временной отметкой. Однако на практике исследователи чаще сталкиваются с более сложными структурами данных — множественными последовательностями событий, связанными с различными пользователями и обладающими уникальными паттернами развития. Основная задача при работе с такими данными заключается в выявлении скрытых закономерностей как на индивидуальном (пользовательском), так и на общем уровне для последующего прогнозирования поведения.

Современные исследования показывают, что в реальных сценариях событие редко может быть адекватно описано единственной меткой. Гораздо более общей и практически значимой является постановка задачи, когда в каждый момент времени может наблюдаться целый набор меток. Типичными примерами служат:

- одновременное использование нескольких сервисов в мобильном приложении
- покупка множества товаров в интернет-магазине
- совершение различных финансовых операций в определенный период

Таким образом, переход от анализа отдельных временных событий к работе с временными наборами представляет собой важное обобщение исходной задачи. Под временными наборами (Temporal Sets) мы понимаем последовательности временно́ размеченных множеств, состоящих из произвольного числа меток. Соответственно, прогнозирование временных наборов формулируется как задача предсказания набора меток для следующего события на основе наблюдаемой последовательности предшествующих наборов.

2.0.1. Формальная постановка задачи

Пусть имеется множество пользователей $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, где N — общее количество пользователей. Для каждого пользователя $u_i \in \mathcal{U}$ задана последовательность временных наборов $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$, где T — количество наблюдаемых временных отметок. Каждый набор s_i^j представляет собой подмножество меток из общего словаря $\mathcal{Y} = \{y_1, y_2, \ldots, y_M\}$ размерности M.

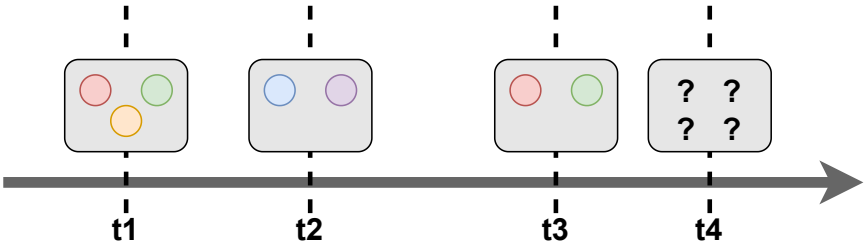

Рис. 2.1: Постановка задачи.

Математически задача формулируется как поиск функции g, которая по последовательности исторических наборов $\{s_i^1, s_i^2, \ldots, s_i^T\}$ предсказывает последующий набор $\hat{s}_i^{T+1}$:

$$\hat{s}_i^{T+1} = g(s_i^1, s_i^2, \ldots, s_i^T, \mathbf{W}) \tag{2.1}$$

где $\mathbf{W}$ — обучаемые параметры модели.

2.1. Определение метрик

Приведем сразу метрики, которые в данной задаче применяемы.

В статье [85] представлен обзор метрик для классификации по нескольким меткам. Приведем набор метрик для оценки качества алгоритмов.

2.1.1. Обозначения

Каждому образцу x_i присваивается набор меток Y_i. Определим:

- $y_{ij} = 1$, если метка j актуальна для экземпляра i, иначе 0
- $\hat{Y}_i$ - предсказанный набор меток
- $g_{ij} = 1$, если модель предсказывает метку j для i, иначе 0
- $f_i(x_j)$ - уверенность модели в метке i для экземпляра j

2.1.2. Основные метрики

- **Micro-F1**: F-мера, усредненная по всей матрице

$$\text{micro-F1} = \frac{2\sum_{j=1}^{K}\sum_{i=1}^{N} y_{ij}g_{ij}}{\sum_{j=1}^{K}\sum_{i=1}^{N}(y_{ij} + g_{ij})}$$

- **Macro-F1**: F-мера, усредненная по классам

$$\text{macro-F1} = \frac{1}{K}\sum_{j=1}^{K}\frac{2\sum_{i=1}^{N} y_{ij}g_{ij}}{\sum_{i=1}^{N}(y_{ij}+g_{ij})}$$

- **Micro-AUC**: Усреднение AUC по матрице

$$\text{micro-AUC} = \frac{|W_{micro}|}{(\sum_{i=1}^{N}|Y_{i\cdot}^{+}|)(\sum_{i=1}^{N}|Y_{i\cdot}^{-}|)}$$
$$W_{micro} = \{(a,b,i,j)|(a,b) \in Y_{\cdot i}^{+} \times Y_{\cdot j}^{-}, f_i(x_a) \geq f_j(x_b)\}$$

- **Macro-AUC**: Усреднение AUC по классам

$$\text{macro-AUC} = \frac{1}{K}\sum_{j=1}^{K}\frac{W_{macro}^{j}}{|Y_{\cdot j}^{+}||Y_{\cdot j}^{-}|}$$
$$W_{macro}^{j} = \{(a,b) \in Y_{\cdot j}^{+} \times Y_{\cdot j}^{-}|f_j(x_a) \geq f_j(x_b)\}$$

- **Hamming Loss**: Доля ошибочных предсказаний

$$\text{HammingLoss} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\mathbb{I}(y_{ij} \neq g_{ij})$$

- **Weighted ROC-AUC**: Взвешенное среднее AUC

$$\text{WeightedAUC} = \frac{\sum_{j=1}^{K} w_j \text{AUC}_j}{\sum_{j=1}^{K} w_j}$$
$$\text{AUC}_j = \frac{1}{|Y_{\cdot j}^{+}||Y_{\cdot j}^{-}|}\sum_{a \in Y_{\cdot j}^{+}}\sum_{b \in Y_{\cdot j}^{-}}\mathbb{I}(f_j(x_a) > f_j(x_b))$$
$$w_j = |Y_{\cdot j}^{+}|$$

2.1.3. Дополнительные метрики

- **Precision@k**: Точность в топ-k

$$\text{Precision@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \hat{Y}_i^{k}|}{k}$$

- **Recall@k**: Полнота в топ-k

$$\text{Recall@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \cap \hat{Y}_i^k|}{|Y_i|}$$

Глава 3
Обзор существующих решений

Постановка задачи классификации с несколькими метками возникает во многих различных областях, например, категоризация текста или разметка изображений, каждая из которых влечет за собой свои особенности и проблемы. Обзор [98] исследует основы обучения с несколькими метками, обсуждая как устоявшиеся методы, так и самые последние подходы. Новые тенденции рассматриваются в более позднем обзоре [42].

Во-первых, мы изучаем функции потерь, адаптированные для настройки с несколькими метками, и некоторые методы составления прогноза набора меток. Во-вторых, мы рассматриваем использование RNN в задаче классификации с несколькими метками. В-третьих, мы рассматриваем, как фиксировать зависимости меток. Затем мы обсуждаем связь с проблемой последовательных рекомендаций. Наконец, мы суммируем анализ литературы и представляем выявленный пробел в исследованиях.

3.0.1. Функции потерь и способы составления набора меток в многометковой задаче.

В статье [50] изучается теоретическая основа для основных подходов к сведению задачи классификации с несколькими метками к серии бинарных или многоклассовых задач. В частности, они показывают, что рассматриваемые сокращения неявно оптимизируются либо для Precision@k, либо для Recall@k. Выбор правильного сокращения должен основываться на конечном показателе производительности, представляющем интерес. В [40] авторы предлагают улучшенную функцию потерь для парного ранжирования в задаче классификации изображений с несколькими метками, которую легче оптимизировать. Кроме того, они обсуждают подход, основанный на оценке оптимальных порогов доверия для части модели, определяющей, какие метки следует включить в окончательный прогноз. Задача классификации текста с несколькими метками является темой [13]. Авторы создают сквозную структуру глубокого обучения под названием ML-Net. ML-Net состоит из двух частей: сети прогнозирования меток и сети прогнозирования количества меток. Чтобы получить окончательный набор меток, ранжируются оценки уверенности, полученные из сети прогнозирования меток, а затем предсказываются верхние K_{top} метки. Отдельная сеть подсчета меток предсказывает K_{top}.

3.0.2. Нейронные сети для предсказания временных наборов меток.

В [90] авторы используют модель RNN для решения проблемы классификации по нескольким меткам. Они решают проблему, связанную с тем, что RNN выдают последовательные выходные данные, поэтому целевые метки должны быть упорядочены. Авторы предлагают динамически упорядочивать метки истинности на основе прогнозов модели, что способствует более быстрому обучению и смягчает эффект дублирования генерации. В свою очередь, [76] рассматривают преобразование проблемы классификации по нескольким меткам в проблему прогнозирования последовательности с помощью декодера RNN. Они предлагают новый алгоритм обучения для декодеров на основе RNN, который не полагается на предопределенный порядок меток. Следовательно, модель исследует различные комбинации меток, смягчая смещение экспозиции. Работа [67] рассматривает ту же постановку задачи классификации по нескольким меткам в потоке событий. Модель авторов нацелена на захват временных и вероятностных зависимостей между типами одновременных событий путем кодирования исторической информации с помощью трансформера и последующего использования условной смеси экспертов Бернулли. В этой статье [93] обсуждается формулировка задачи прогнозирования временных наборов для пользователей. Она предлагает непрерывную систему обучения, которая позволяет явно фиксировать изменяющиеся предпочтения пользователей, поддерживая банк памяти, который может хранить состояния всех пользователей и элементов. В этой парадигме авторы конструируют неубывающую универсальную последовательность, содержащую все определяемые пользователем взаимодействия, а затем хронологически изучают каждое взаимодействие. Для исследования перекрестной связи между продуктами в корзине был предложен ConvTSP [97], который объединяет динамические интересы пользователя и статистические интересы в единое векторное представление для пользователя.

3.0.3. Подходы к использованию зависимостей меток.

Авторы работы [35] разработали модель C-Tran для многоклассовой классификации изображений, основанную на архитектуре трансформера. Основная инновация заключается в использовании маскирования меток при обучении, что позволяет эффективно выявлять взаимосвязи между признаками изображений и целевыми метками.

В исследовании [91] предложена глубокая нейросетевая архитектура, которая создает совместные эмбеддинги для признаков и меток, учитывая их взаимозависимости. Для повышения эффективности модели

авторы разработали специальную функцию потерь, чувствительную к корреляции между метками.

Среди современных подходов к учету взаимосвязей между метками особое внимание заслуживает применение графовых нейронных сетей. В частности, в работе [54] для задачи многоклассовой классификации текстов используется механизм Graph Attention Network (GAT). Предложенная архитектура объединяет векторные представления из BiLSTM с взвешенными признаками меток, полученными через GAT, для формирования итогового набора предсказанных меток.

Аналогичные задачи по выявлению зависимостей решаются и при обработке последовательностей событий, где применяются специализированные механизмы внимания [49]. Наиболее близкой к нашему подходу LANET является работа [41], в которой исследуются взаимосвязи между временными рядами для многомерной классификации. Авторы используют двунаправленный механизм внимания (по временным шагам и каналам) для генерации эмбеддингов, которые затем подаются на классификатор.

3.0.4. Системы последовательной рекомендации.

Другим близким соседом нашей постановки проблемы является проблема построения последовательной рекомендательной системы [81, 56]. В этом случае у нас есть много возможных меток, и мы должны отсортировать их по вероятности появления в следующий раз. Обычно оценка векторных представлений для всех возможных меток/элементов является частью конвейера. Существующие подходы используют нейронные сети для последовательных данных, таких как LSTM [83], а также механизм внимания [80]. Мы хотим выделить утверждение, связанное с использованием только недавних прошлых данных для прогнозирования [38]. Однако миллионы возможных меток обычно приводят к более классическим методам в этой области с конкретными функциями и методами потерь.

3.0.5. Рекомендация корзины в следующий момент времени.

В этом разделе мы представим статьи, связанные с проблемой рекомендации следующей корзины. Эта формулировка похожа на нашу, поэтому мы также рассмотрели множество подходов и идей при анализе нашей области исследований. Авторы в [3] предложили персонализированную модель, которая фиксирует краткосрочные зависимости в рамках временного набора продуктов, а также долгосрочную на основе исторической информации о пользователях. Также в [89] для связи локальной и глобальной информации о пользователях предлагается ги-

бридный метод на основе автокодировщика для извлечения контекста и рекуррентная нейронная сеть для понимания динамики изменения интересов. Для преодоления подобных проблем создается сеть внимания на основе гипер-ребер на основе графа [69] для следующей рекомендации. В этой формулировке проблемы возникает сложность работы со словарем категорий продуктов, поскольку они насчитывают тысячи значений, [78] использует GRU для прогнозирования следующей корзины, что легко масштабируется до большого ассортимента.

3.1. Временные точечные процессы

Здесь мы приводим более подробную информацию о наиболее важных работах, чтобы поместить нашу работу в контекст. Данное агрегирование будет с теоретической точки зрения будет применимо к данным процессам.

Временные точечные процессы (TPP) Для *временного* точечного процесса каждая точка лежит на временной оси $\mathbb{R}^1$. Следовательно, с каждой точкой связано время. Для *отмеченного* временного точечного процесса у нас есть дополнительная метка или вектор признаков, связанных с каждой точкой. Можно сказать, что вектор признаков принадлежит $\mathbb{R}^d$ [88]. Таким образом, последовательности событий являются частным случаем реализаций отмеченных временных точечных процессов. Самовозбуждающиеся точечные процессы представляют особый интерес, поскольку они предполагают, что будущая интенсивность событий зависит от истории. Обычно прошлые события увеличивают интенсивность в будущем — например, это касается ретвитов в социальных сетях [58] или распространения вируса [8].

Глубокое обучение для TPP В то время как классическое машинное обучение успешно моделирует сложную динамику последовательностей событий [103], внедрение глубокого обучения приводит к большей адаптивности и, следовательно, лучшим результатам [47]. Улучшение достигается за счет гибких глубоких нейронных сетей, которые могут обрабатывать последовательности произвольной сложности, если обучающая выборка достаточно велика. Более того, возможности представления возникают, поскольку нейронная сеть может выводить представления либо для каждого момента времени, события, либо последовательности событий в целом, что позволяет решать другие проблемы нисходящего потока, используя эти представления. Напротив, основным недостатком глубокого обучения являются более высокие вычислительные затраты на обучение таких моделей [104].

Разнообразие архитектур глубокого обучения приводит к множеству способов моделирования временных точечных процессов. Исторически первые статьи рассматривали различные рекуррентные архитектуры, такие как непрерывная LSTM [47] или RMTPP [14]. Адаптация архитектур трансформаторов последовала через, например, [106, 99]. Сверточная нейронная сеть рассматривалась непосредственно в [66], где авторы достигли многообещающих результатов. Однако их архитектура имеет только два слоя, и только первый из них отвечает за изменение времени с неоднородного на однородное. Статьи о слоях пространства состояний [19] также упоминают неоднородность времени как одну из проблем, не предоставляя рецепта для эффективного слоя пространства состояний с неоднородным временем.

Мы также отмечаем, что обучение на основе TPP происходит самоконтролируемым способом без дополнительной информации о метках, поскольку мы можем получить типы событий и время возникновения событий из самих данных. Однако прямые подходы к обработке последовательных данных самоконтролируемым способом для временных рядов и последовательностей событий на основе контрастных [95, 4], неконтрастных [46] и генеративных стратегий [29] являются дополнительными к TPP.

Глава 4
Теория

Задача прогнозирования временных множеств была формализована следующим образом. Пусть $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ — набор из N пользователей. Каждый пользователь $i, 1 \leq i \leq N$, связан с последовательностью временных множеств $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$, где T — количество наблюдаемых временных меток. Набор $s_i^j, 1 \leq i \leq N, 1 \leq j \leq T$, — это набор произвольного количества меток, выбранных из словаря $\mathcal{Y} = \{y_1, y_2, \ldots, y_M\}$ размера M. Учитывая последовательность исторических наборов $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ для пользователя $u_i \in \mathcal{U}$, где каждый набор $s_i^j \subset \mathcal{Y}$, цель задачи прогнозирования временных наборов состоит в прогнозировании последующего набора меток $\hat{s}_i^{T+1}$, то есть,

$$\hat{s}_i^{T+1} = g(s_i^1, s_i^2, \ldots, s_i^T, \mathbf{W}),$$

где $\mathbf{W}$ относится к обучаемым параметрам функции g. Функция g должна уметь улавливать последовательное развитие наборов в последовательности $\mathcal{S}_i$, а также взаимодействие меток внутри каждого набора s_i^j.

4.1. Подход LANET

Основными аспектами проблемы прогнозирования временных множеств являются эволюционирующая во времени природа серий множеств и сложная внутренняя организация отдельных множеств. Примечательно, что эти особенности взаимосвязаны и дополняют друг друга, требуя совместной записи. Помня о важности их одновременной обработки, мы предлагаем модель LANET, которая нацелена на такую задачу. В частности, мы предлагаем вычислять внутреннее внимание между специально разработанными представлениями исторической информации. Такие представления охватывают знание времени происходящих событий и состав меток каждого набора, связанного с событием. Использование механизма внутреннего внимания над сконструированными представлениями позволяет идентифицировать отношения, учитывающие время и метки. Наконец, мы применяем аффинные преобразования к обновленным представлениям на выходе внутреннего внимания, чтобы получить вектор оценок уверенности для следующих меток событий.

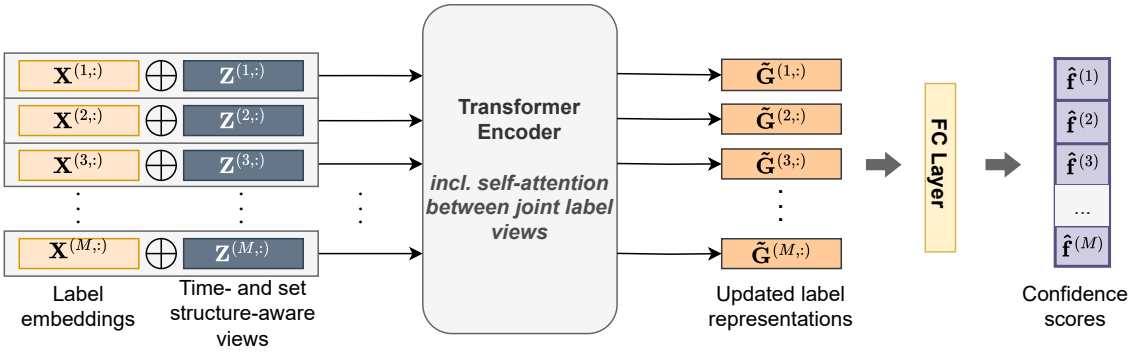

Рис. 4.1: Архитектура LANET для прогнозирования временных наборов. Ключевая часть — агрегация исторической информации в репрезентативные представления, которые будут переданы в блок энкодера трансформера. Выход модели — вектор оценок уверенности, компоненты которого связаны с перспективой того, что соответствующая метка будет членом набора следующего события.

4.1.1. Представление исторической информации в LANET

Прежде всего, мы хотим эффективно агрегировать прошлую информацию о времени событий и задать структуры для последовательности $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$. Пусть $\mathbf{X} \in \mathbb{R}^{M \times D}$ обозначает матрицу представлений всех меток из словаря $\mathcal{Y} = \{y_1, y_2, \ldots, y_M\}$, где D — размерность векторных представлений. Параметры матрицы $\mathbf{X}$ инициализируются из стандартного нормального распределения и позже обновляются в процессе обучения. Важным шагом является построение представлений времени. Каждый набор s_i^j связан со временем t_j. Отсчет времени начинается с одной общей точки для всех пользователей. Для каждой временной метки $j, 1 \leq j \leq T$, мы устанавливаем временное векторное представление $\mathbf{t}_j \in \mathbb{R}^D$, как это делается в [67]:

$$\mathbf{t}_j^{(d)} = \begin{cases} \cos\left(t_j / 10000^{\frac{d-1}{D}}\right), & \text{если } d \text{ является нечетным,} \\ \sin\left(t_j / 10000^{\frac{d}{D}}\right), & \text{если } d \text{ является четным,} \end{cases}$$

где $d, 1 \leq d \leq D$, является компонентой вектора размерности D. После определения представления для каждого момента времени $t_j, 1 \leq j \leq T$, мы агрегируем все знания, связанные со временем, из последовательности $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ в матрицу $\mathbf{Z} \in \mathbb{R}^{M \times D}$. m-я строка, $1 \leq m \leq M$, матрицы $\mathbf{Z}$, обозначенная как $\mathbf{Z}^{(m,:)}$, равна сумме эмбеддингов временных меток, в которой метка $y_m \in \mathcal{Y}$ появляется как член множества:

$$\mathbf{Z}^{(m,:)} = \sum_{j | y_m \in s_i^j} \mathbf{t}_j$$

Если метка y_m не встречается ни в одном наборе последовательности $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$, то m-я строка матрицы $\mathbf{Z}$ будет состоять из

одних нулей. Следовательно, в случае встречи метки y_m в нескольких наборах последовательности $\mathcal{S}_i$, соответствующая m-я строка матрицы $\mathbf{Z}$ будет суммой всех соответствующих временных эмбеддингов для этой конкретной метки.

Объединенное представление последовательности $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ представляет собой конкатенацию определенных матриц, воплощающих информацию о времени и структуре множества:

$$\mathbf{G} = \mathbf{X} \oplus \mathbf{Z}$$

Строки результирующей матрицы $\mathbf{G} \in \mathbb{R}^{M \times 2D}$ рассматриваются как совместные представления соответствующих меток. А именно, m-я строка матрицы $\mathbf{G}$ представляет собой совместный вид метки y_m. Разработанное представление каждой метки включает ее вид, выраженный в $\mathbf{X}$, и часть, отвечающую за взаимодействие с другими метками с учетом времени, находящимися в $\mathbf{Z}$.

4.1.2. Изучение отношений посредством собственного внимания в энкодере LANET

Мы определяем совместные представления меток как строки матрицы $\mathbf{G}$, которые включают в себя самоориентированную информацию меток, а также знание взаимосвязей меток с учетом времени. Для поощрения дальнейшего захвата отношений мы применяем механизм самовнимания к матрице $\mathbf{G}$, чтобы получить ее обновленную версию $\tilde{\mathbf{G}}$:

$$\tilde{\mathbf{G}} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\mathbf{T}}{\sqrt{2D}})\mathbf{V},$$

где $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ — матрицы запроса, ключа и значения, которые являются линейными преобразованиями матрицы $\mathbf{G}$. Основной блок архитектуры LANET состоит из нескольких слоев энкодера с многоголовым вниманием. Используя его, идет объединение исторических записей, выраженных через представления с совместными метками, и подчеркиваются существенные взаимодействия. Обновленные представления меток наполнены ретроспективной информацией о времени и структуре набора.

4.1.3. Слой предсказания

Наконец, обновленные представления $\tilde{\mathbf{G}} \in \mathbb{R}^{M \times 2D}$ принимают участие в получении оценок достоверности для всех меток, которые будут включены в набор следующего события:

$$\hat{\mathbf{f}} = \text{sigmoid}(\tilde{\mathbf{G}}\mathbf{W}^{\text{out}} + b^{\text{out}}),$$

где $\hat{\mathbf{f}} \in \mathbb{R}^M$ — вектор оценок достоверности размером со словарь меток, $\mathbf{W}^{\text{out}} \in \mathbb{R}^{2D \times 1}$ и $b^{\text{out}} \in \mathbb{R}$ — обучаемые параметры слой прогнозирования. Используется сигмоидальная функция активации, чтобы оценки достоверности лежали в диапазоне $[0, 1]$. Таким образом, m-й компонент, $1 \leq m \leq M$, вектора достоверности $\hat{\mathbf{f}} \in \mathbb{R}^M$ связан с перспективой метки y_m стать частью прогнозируемого набора $\hat{s}_i^{T+1}$.

4.1.4. Процесс обучения модели

Выход слоя прогнозирования LANET — вектор оценок достоверности $\hat{\mathbf{f}} \in \mathbb{R}^M$. Вектор $\hat{\mathbf{f}}$ обеспечивает основу для прогнозирования состава на наборе следующего события $\hat{s}_i^{T+1}$. Для обучения и проверки модели используется реальный следующий набор s_i^{T+1} в качестве базовой истины. LANET обучается сквозным образом, принимая историческую последовательность $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ в качестве входных данных и создавая вектор оценки уверенности $\hat{\mathbf{f}}$ в качестве выходных данных. Принимаем следующую функцию потерь:

$$\mathcal{L}_i = -\frac{1}{M} \sum_{m=1}^{M} \left(\mathbf{I}_m \log \hat{\mathbf{f}}^{(m)} + \mathbf{I}_m^{'} \log \left(1 - \hat{\mathbf{f}}^{(m)}\right) \right),$$

где $\mathbf{I}_m = \mathbf{I}\{y_m \in s_i^{T+1}\}$ — индикаторная функция метки y_m, которая является членом множества s_i^{T+1}, тогда как $\mathbf{I}_m^{'}$ — индикаторная функция с противоположным условием $\mathbf{I}_m^{'} = \mathbf{I}\{y_m \notin s_i^{T+1}\}$. Обозначим m-й компонент вектора прогнозируемой уверенности $\hat{\mathbf{f}}$ как $\hat{\mathbf{f}}^{(m)}$. Формула функции потерь $\mathcal{L}_i$ приведена для случая, когда мы рассматриваем только одного пользователя u_i. Учитывая всех доступных пользователей $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, мы минимизируем сумму всех компонентов потерь, связанных с пользователями $\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_i$ в процессе обучения.

В обучающем наборе данных для LANET связка набора s_i^{T+1} в качестве истинного значения и последовательности $\{s_i^1, s_i^2, \ldots, s_i^T\}$ в качестве входных данных — это не единственный обучающий пример, который извлекается из пользовательской последовательности $\mathcal{S}_i$. Чтобы увеличить объем обучающих данных, мы также используем все промежуточные наборы в последовательности в качестве истинного значения и предыдущие наборы в качестве входных данных модели. Таким образом, $s_i^j, 2 \leq j \leq T+1$, берутся в качестве целевых наборов, а подпоследовательности $\{s_i^1, \ldots, s_i^{(j-1)}\}$ — в качестве соответствующих входных данных.

4.2. Эффективность LANET

Определение 1. Временная последовательность множеств Пусть $\mathcal{Y} = \{y_1, y_2, ..., y_M\}$ — множество категориальных меток, $|\mathcal{Y}| = M$. Временная последовательность множеств — это последовательность $\mathcal{S} = (s_1, s_2, ..., s_T)$, где $s_t \subseteq \mathcal{Y}$, $\forall t \in [1,T]$, и каждое событие s_t сопоставлено временной отметке $t_t \in \mathbb{R}$.

Определение 2. Представление меток Каждой метке $y_m \in \mathcal{Y}$ ставится в соответствие обучаемый эмбеддинг $\mathbf{x}_m \in \mathbb{R}^D$, где D — размерность пространства представлений.

Матрица меток:

$$X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M)^\top \in \mathbb{R}^{M \times D}.$$

Определение 3. Агрегация по меткам Для метки y_m определим:

$$N_m = \sum_{t=1}^{T} \chi_m(t), \quad \text{где } \chi_m(t) = \mathbf{1}(y_m \in s_t),$$

тогда агрегированный эмбеддинг метки:

$$\mathbf{Z}_{\text{label}}(m, :) = N_m \cdot \mathbf{x}_m.$$

Объединённое представление:

$$G_{\text{label}} = X \oplus Z_{\text{label}} \in \mathbb{R}^{M \times 2D}.$$

Определение 4. Временная агрегация Для метки y_m:

$$\mathbf{Z}_{\text{time}}(m, :) = \sum_{t:\, y_m \in s_t} \mathbf{t}_t,$$

где $\mathbf{t}_t \in \mathbb{R}^D$ — временной эмбеддинг события s_t.

Объединённое представление:

$$G_{\text{time}} = X \oplus Z_{\text{time}} \in \mathbb{R}^{M \times 2D}.$$

Теорема Если $M \ll T$, то время выполнения трансформерного слоя внимания при использовании агрегации по меткам удовлетворяет неравенству:

$$T_{\text{label}} \ll T_{\text{time}},$$

где:

$$T_{\text{label}} = O(M^2 D), \quad T_{\text{time}} = O(T^2 D).$$

Лемма 1. Сложность вычисления матрицы внимания Для входной матрицы $G \in \mathbb{R}^{L \times D}$, где L — число элементов, время вычисления матрицы внимания:

$$T_{\text{attn}} = O(L^2 D).$$

Лемма 2. Сравнение сложностей Если $M \ll T$, то отношение сложностей:

$$\frac{T_{\text{label}}}{T_{\text{time}}} = O\left(\frac{M^2}{T^2}\right) \to 0, \quad T \to \infty.$$

Доказательство

Шаг 1. Вычислительная сложность трансформерного слоя

Сложность вычисления матрицы внимания:

$$T_{\text{attn}} = O(L^2 D),$$

где L — количество обрабатываемых объектов (меток или временных точек).

Шаг 2. Применение к случаям агрегаций

- При меточной агрегации: $L = M$, поэтому:

$$T_{\text{label}} = O(M^2 D).$$

- При временной агрегации: $L = T$, поэтому:

$$T_{\text{time}} = O(T^2 D).$$

Шаг 3. Сравнение при условии $M \ll T$

По условию $M \ll T$, т.е. $\exists c > 0$, такое что:

$$\lim_{T \to \infty} \frac{M}{T} = 0.$$

Тогда:

$$\frac{T_{\text{label}}}{T_{\text{time}}} = \frac{O(M^2 D)}{O(T^2 D)} = O\left(\frac{M^2}{T^2}\right) \to 0.$$

Следовательно:

$$T_{\text{label}} \ll T_{\text{time}}.$$

Теорема (Оптимальная агрегация меток в процессах Хокса) Пусть $(\Omega, \mathcal{F}, P)$ — вероятностное пространство. Рассмотрим многомерный процесс Хокса $\{N_t^m\}_{t \geq 0}$, где $m = 1, ..., M$ обозначает тип события (метку), и пусть интенсивность для типа m задаётся как:

$$\lambda_t^m = \mu_m + \sum_{n=1}^{M} \int_0^t g_{mn}(t-s) dN_s^n,$$

где $\mu_m > 0$ — базовая интенсивность, $g_{mn} : \mathbb{R}_+ \to \mathbb{R}_+$ — ядро возбуждения между событиями типов m и n, удовлетворяющее условиям:

$$\sup_{m,n} \int_0^\infty g_{mn}(u)du < \infty.$$

Обозначим через $\mathcal{H}_t$ историю всех событий до времени t, то есть $\mathcal{H}_t = \sigma(N_s^m : 0 \leq s \leq t, m = 1, ..., M)$.

Пусть также заданы функции потерь $\ell : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}_+$, измеряющие отклонение предсказанных интенсивностей от истинных.

Рассмотрим агрегированный по меткам сигнал:

$$S_t^m = \frac{1}{t} \int_0^t \phi(m, N_s)ds,$$

где $\phi(m, N_s)$ — некоторая функция, зависящая от истории событий по метке m к моменту s.

Утверждение Если $\phi(m, N_s)$ является достаточной статистикой для λ_t^m при фиксированной временной динамике, то существует последовательность оценок $\{\hat{\lambda}_t^m\}_{t\geq 0}$, основанных на $\{S_t^m\}_{t\geq 0}$, такая что:

$$\lim_{t\to\infty} \mathbb{E}\left[\ell(\hat{\lambda}_t^m, \lambda_t^m)\right] = 0,$$

и эта сходимость равномерна по $m = 1, ..., M$.

Доказательство

Шаг 1: Существование устойчивого представления

По условию теоремы, ядра g_{mn} имеют конечные интегралы:

$$\int_0^\infty g_{mn}(u)du < C < \infty.$$

Следовательно, матрица $\Gamma = (\gamma_{mn})$, где $\gamma_{mn} = \int_0^\infty g_{mn}(u)du$, имеет спектральный радиус $\rho(\Gamma) < 1$, что гарантирует существование стационарного режима для процесса Хокса.

В этом случае можно определить эмпирическое распределение частот меток:

$$\bar{N}_t^m = \frac{1}{t} N_t^m.$$

Известно, что при условиях регулярности:

$$\bar{N}_t^m \xrightarrow{a.s.} \nu_m, \quad t \to \infty,$$

где $\nu_m = \mathbb{E}[N_t^m]/t$ — стационарная интенсивность для метки m.

Шаг 2: Агрегация по меткам

Рассмотрим агрегированный сигнал:

$$S_t^m = \frac{1}{t}\int_0^t \phi(m, N_s)ds.$$

Предположим, что $\phi(m, N_s)$ зависит только от числа событий метки m до момента s, т.е.

$$\phi(m, N_s) = f\left(\frac{d}{ds}N_s^m\right),$$

где f — гладкая функция.

Тогда:

$$S_t^m = \frac{1}{t}\int_0^t f\left(\frac{d}{ds}N_s^m\right)ds.$$

Заметим, что $\frac{d}{ds}N_s^m = \sum_{k:T_k^m \le s}\delta(s - T_k^m)$, где T_k^m — моменты событий метки m. Тогда:

$$S_t^m = \frac{1}{t}\sum_{k:T_k^m \le t} f(1),$$

поскольку производная скачка равна 1.

Отсюда:

$$S_t^m = \frac{N_t^m}{t}f(1).$$

Таким образом, S_t^m содержит информацию о плотности событий метки m.

Шаг 3: Предсказание интенсивности

Пусть $\hat{\lambda}_t^m = h(S_t^m)$, где h — подходящая функция. Тогда:

$$\hat{\lambda}_t^m = h\left(\frac{N_t^m}{t}f(1)\right).$$

Из закона больших чисел:

$$\frac{N_t^m}{t} \xrightarrow{a.s.} \nu_m.$$

Следовательно:

$$\hat{\lambda}_t^m \xrightarrow{a.s.} h(\nu_m f(1)).$$

Но известно, что $\nu_m = \mathbb{E}[\lambda_t^m]$. Значит, если выбрать h так, чтобы $h(\nu_m f(1)) = \nu_m$, то:

$$\hat{\lambda}_t^m \xrightarrow{a.s.} \mathbb{E}[\lambda_t^m].$$

Шаг 4: Минимизация функции потерь

Пусть $\ell(x, y) = \|x - y\|^2$. Тогда:

$$\mathbb{E}[\ell(\hat{\lambda}_t^m, \lambda_t^m)] = \mathbb{E}[\|\hat{\lambda}_t^m - \lambda_t^m\|^2].$$

Подставляя $\hat{\lambda}_t^m = h(S_t^m)$, получаем:

$$\mathbb{E}[\|\hat{\lambda}_t^m - \lambda_t^m\|^2] = \mathbb{E}[\|h(S_t^m) - \lambda_t^m\|^2].$$

Поскольку $S_t^m \to \nu_m f(1)$ п.н., то:

$$\|h(S_t^m) - \lambda_t^m\| \xrightarrow{a.s.} 0.$$

По теореме Лебега о мажорируемой сходимости:

$$\mathbb{E}[\|h(S_t^m) - \lambda_t^m\|^2] \to 0.$$

Шаг 5: Равномерная сходимость по меткам
Поскольку $\sup_m \nu_m < \infty$, и функция h не зависит от m, то:

$$\sup_{m=1,\dots,M} \mathbb{E}[\|h(S_t^m) - \lambda_t^m\|^2] \to 0.$$

Таким образом, сходимость равномерна по m.

Вывод Существует способ агрегации информации по меткам, который позволяет строить состоятельные оценки интенсивностей в процессах Хокса, минимизируя функцию потерь в среднем квадратичном смысле. Это подтверждает эффективность агрегации по меткам вместо временного усреднения.

■

4.3. Выводы по разделу

В заключение, представленный подход LANET предлагает эффективное решение для прогнозирования временных множеств, объединяя временные и структурные особенности данных. Модель демонстрирует способность улавливать сложные зависимости между метками, учитывая как их совместное появление, так и временную динамику. Использование механизма самовнимания позволяет выявлять значимые взаимодействия между метками, что повышает точность прогнозирования. Предложенный метод агрегации временных эмбеддингов обеспечивает учет исторической информации в компактной и интерпретируемой форме. Слой предсказания на основе сигмоидальной функции позволяет получать вероятностные оценки включения меток в будущие наборы. Обучение модели с использованием бинарной кросс-энтропии показало свою эффективность для задач такого типа.

Глава 5
Описание практической части

В данной главе описываются технические аспекты выполненной работы.

5.1. Данные

5.1.1. Описание данных

В этом разделе мы представляем сравнение производительности нашего подхода LANET с существующими моделями для задачи прогнозирования временных множеств. Кроме того, мы проводим тщательное исследование абляции, которое раскрывает понимание рабочих деталей LANET.

Проведя анализ работ, посвященных моделям прогнозирования временных множеств, мы выделяем четыре часто используемых набора данных:

- **Dunnhumby-Carbo (DC)** [15]: Этот набор данных включает транзакционные данные домохозяйств в розничном магазине за два года. Здесь наборы — это продукты, назначенные для одной транзакции.

- **Mimic III** [51]: Он состоит из медицинских карт пациентов из отделения интенсивной терапии. Событие, связанное с пациентом, представляет собой время госпитализации и набор кодов классификации заболеваний.

- **Instacar**t [25]: Набор данных Instacart содержит записи заказов продуктов пользователями. Каждое событие описывается временем покупки и набором меток продуктов.

- **Synthea** [53]: Это искусственно созданные данные EHR с имитированными медицинскими событиями, аналогичные набору данных MIMIC III.

Статистика этих наборов данных приведена в таблице 5.1. Мы предоставляем общее количество наборов в каждом наборе данных (#Sets), медианный размер набора (MdnSS), максимальный размер набора (MaxSS), размер словаря меток (Vocab), среднюю длину исторических последовательностей (MnLen) и количество доступных последовательностей (#Seqs).

Таблица 5.1: Statistics of the datasets for temporal sets prediction.

Dataset	#Sets	MdnSS	MaxSS	Vocab	MnLen	#Seqs
Synthea	108 439	2	13	232	44.1	2459
Mimic III	17 849	5	23	169	2.7	6636
Synthea	108 439	2	13	232	44.1	2459
DC	121 165	1	9	217	3.6	33895
Instacart	115 604	6	43	134	16.5	7000

5.2. Методы

В данном разделе рассмотрим методы, с которыми проводилось сравнения на наборах данных выше:

5.2.1. GPTopFreq

Данный подход это базовый метод на основе частоты, вдохновленный [39]. Он оценивает частоты появления каждой метки во всем наборе данных и в истории, связанной с пользователем. Затем для каждой метки GPTopFreq берет максимальную из “общих” и “личных"частот и использует ее как прогнозируемую вероятность.

5.2.2. DNNTSP

Подход [101] DNNTSP (Deep Neural Network for Time Series Prediction) представляет собой метод прогнозирования временных рядов на основе глубоких нейронных сетей. Авторы предлагают архитектуру, сочетающую сверточные и рекуррентные слои для эффективного извлечения временных зависимостей. Модель использует механизм внимания для выделения наиболее значимых временных шагов и улучшения точности прогноза. Обучение проводится на мини-батчах с оптимизацией через Adam для ускоренной сходимости. Для борьбы с переобучением применяются dropout и L2-регуляризация. Входные данные предварительно нормализуются, а окна временных рядов формируются с учетом временных лагов. Эксперименты показали, что DNNTSP превосходит традиционные методы (ARIMA, LSTM) на разнообразных датасетах. Ключевым преимуществом подхода является его адаптивность к нелинейным и нестационарным временным рядам. Авторы также предлагают модификацию для многомерного прогнозирования с учетом взаимосвязей между признаками. Результаты демонстрируют высокую точность как на коротких, так и на длинных горизонтах прогнозирования. Реализация данного подхода была взята тут[1].

[1] https://github.com/yule-BUAA/DNNTSP

5.2.3. SFCNTSP

Подход в [94] SFCNTSP (Spatial-Feature Convolutional Neural Network for Time Series Prediction) — это метод прогнозирования временных рядов, основанный на использовании сверточных нейронных сетей (CNN) с акцентом на пространственно-временные зависимости. В отличие от традиционных CNN, SFCNTSP учитывает не только временные, но и пространственные взаимосвязи в данных, что особенно полезно для многомерных временных рядов. Модель применяет несколько слоев одномерных сверток для извлечения локальных паттернов, а затем объединяет их с глобальными зависимостями через механизмы пулинга и полносвязные слои. Для улучшения устойчивости к шумам используется аугментация данных и батч-нормализация. Архитектура также включает skip-connections, чтобы избежать проблемы исчезающих градиентов при обучении глубоких сетей. Эксперименты показывают, что SFCNTSP эффективно работает в задачах прогнозирования с высокой нелинейностью, превосходя стандартные CNN и RNN-модели. Важным преимуществом подхода является его способность автоматически выделять значимые признаки без ручного проектирования. Авторы также демонстрируют применимость метода в реальных сценариях, таких как финансовая аналитика и прогнозирование энергопотребления. Результаты подтверждают, что SFCNTSP обеспечивает высокую точность даже при наличии пропущенных данных и шумов. Реализация данного подхода была взята тут [2].

5.2.4. TCMBN

TCMBN [67] (Temporal Convolutional Mixture Bayesian Network) — это гибридный метод прогнозирования временных рядов, сочетающий временные сверточные сети (TCN) с байесовскими смешанными моделями для учета неопределенности в данных. Основная идея заключается в использовании TCN для извлечения сложных временных зависимостей, а затем применения байесовской сети для моделирования вероятностных распределений и многомодальности в прогнозах. Архитектура включает dilated-свертки, позволяющие улавливать долгосрочные зависимости, и механизм внимания для выделения ключевых временных интервалов. Байесовский подход позволяет оценивать доверительные интервалы прогнозов, что особенно важно в задачах с высокой волатильностью, таких как финансы или энергетика. Для обучения используется вариационный вывод, что делает метод вычислительно эффективным даже для больших объемов данных. TCMBN демонстрирует устойчивость к шумам и пропускам в данных благодаря вероятностной природе модели. Экспе-

[2] https://github.com/yule-BUAA/SFCNTSP

рименты показывают его превосходство над чисто детерминированными подходами (например, LSTM, TCN) в задачах, где важна оценка неопределенности. Ключевое преимущество — способность выдавать не только точечные прогнозы, но и распределения, что полезно для риск-менеджмента. Метод также поддерживает многомерное прогнозирование с учетом взаимных зависимостей между переменными. Результаты подтверждают его эффективность в реальных приложениях, таких как прогнозирование спроса или медицинских показателей. Реализация данного подхода была взята тут[3].

Мы берем эти модели, поскольку они довольно недавние в прогнозировании временных наборов и демонстрируют высокую производительность. Их гиперпараметры для разных наборов данных установлены на значения, указанные авторами.

5.3. Детали реализации

Наша модель LANET состоит из нескольких слоев энкодера с многоголовым самовниманием. Количество слоев во всех случаях равно 2, а количество голов самовнимания варьируется от 4 до 6 в зависимости от конкретного набора данных. За основу мы берем реализацию слоя энкодера из PyTorch [1]. Мы применяем dropout с вероятностью 0.2 непосредственно к выходу блока энкодера. Зависимость качества LANET от гиперпараметров модели будет представлена в разделе 6. Для процедуры обучения мы используем оптимизатор Adam с начальной скоростью обучения 0.001. Для планировщика мы принимаем стратегию “reduce on Plateau” с терпением 10 эпох и коэффициентом 0.9.

Исходные наборы данных делятся на обучающие, проверочные и тестовые наборы. Разделения выполняются по пользователям. Таким образом, периоды времени в обучающей, валидной и тестовой частях перекрываются. Мы берем 60% выборок данных для обучения модели, 20% для проверки и 20% для тестирования. Все эксперименты запускаются с пятью различными случайными начальными числами; вычисляются среднее и стандартное отклонение результатов.

Оценка прогнозирования временных наборов аналогична проверке многомаркерной классификации, поэтому мы используем общепризнанные метрики из [85] и метрики из соответствующих работ [67] рассматриваемой области временных наборов. Таким образом, мы используем Hamming Loss, Weighted ROC-AUC, Weighted F1, Метрики Micro-F1 и Macro-F1 для окончательной оценки качества. Между тем, расчет micro-F1, macro-F1 и Weighted F1 подразумевает работу с прогнозируемыми наборами меток, а не с оценками достоверности меток. В связи с этим переход

[3] https://github.com/xshou1990/TCMBN

от выходных оценок к прогнозируемым наборам меток осуществляется путем сравнения оценок достоверности, связанных с метками, с определенными пороговыми значениями. Эти пороговые значения рассчитываются на проверочном наборе путем оптимизации оценки F1 для каждой метки отдельно.

Глава 6
Результаты

Таблица 6.1: Сравнение подхода our LANET с существующими моделями для прогнозирования временных наборов на основе четырех наборов данных. Выделены - топ 1, подчеркнуты - топ 2.

Dataset	Model	Micro F1↑	Macro F1↑	Weighted F1↑	Weighted ROC-AUC↑	Hamming Loss↓
Synthea	SFCNTSP	0.2369 ± 0.0156	0.0587 ± 0.0069	0.1656 ± 0.0194	0.6655 ± 0.0077	0.0212 ± 0.0005
	DNNTSP	0.3893 ± 0.0181	0.1288 ± 0.0058	0.2982 ± 0.0132	0.7070 ± 0.0076	0.0183 ± 0.0006
	GPTopFreq	0.4100 ± 0.0042	0.1312 ± 0.0097	0.3286 ± 0.0083	0.7229 ± 0.0093	0.0183 ± 0.0003
	TCMBN	0.4551 ±0.0126	0.1522 ± 0.0023	0.3538 ± 0.0080	0.8347 ± 0.0047	**0.0173 ± 0.0004**
	LANET(ours)	**0.5277 ± 0.0098**	**0.2724 ± 0.0122**	**0.4704 ± 0.0071**	**0.9026 ± 0.0018**	0.0175 ± 0.0005
Mimic III	SFCNTSP	0.4298 ±0.0032	0.2338 ± 0.0071	0.3791 ± 0.0081	0.7034 ± 0.0024	0.0377 ± 0.0004
	DNNTSP	0.4362 ±0.0025	0.2552 ± 0.0034	0.3928 ± 0.0030	0.6926 ± 0.0003	0.0365 ± 0.0003
	GPTopFreq	0.4405 ±0.0070	0.3089 ± 0.0039	0.4291 ± 0.0073	0.6912 ± 0.0028	0.0398 ± 0.0005
	TCMBN	0.5419 ±0.0151	0.2603 ± 0.0276	0.4979 ± 0.0180	0.8670 ± 0.0095	0.0305 ± 0.0008
	LANET(ours)	**0.8218± 0.0211**	**0.7408 ± 0.0377**	**0.8214 ± 0.0224**	**0.9852 ± 0.0023**	**0.0220 ± 0.0001**
DC	SFCNTSP	0.1081 ±0.0058	0.0831 ± 0.0047	0.0886 ± 0.0054	0.7014 ± 0.0024	0.0077 ± 0.0001
	DNNTSP	0.0356 ± 0.0041	0.0254 ± 0.0031	0.0259 ± 0.0027	0.6784 ± 0.0000	0.0074 ± 0.0000
	GPTopFreq	0.1623 ± 0.0019	0.1449 ± 0.0027	0.1525 ± 0.0019	0.6533 ± 0.0022	0.0083 ± 0.0001
	TCMBN	0.2288 ± 0.0153	0.1788 ± 0.0136	0.1968 ± 0.0134	0.8932 ± 0.0048	**0.0073 ± 0.0001**
	LANET(ours)	**0.5608 ± 0.0097**	**0.5473 ± 0.0134**	**0.5498 ± 0.0137**	**0.9941 ± 0.0004**	0.0085 ± 0.0002
Instacart	SFCNTSP	0.2756 ± 0.0140	0.0283 ± 0.0031	0.1672 ± 0.0112	0.6852 ± 0.0448	0.0581 ± 0.0004
	DNNTSP	0.4476 ±0.0021	0.2623 ± 0.0041	0.4160 ± 0.0009	0.7913 ± 0.0004	0.0541 ± 0.0002
	GPTopFreq	0.4376 ±0.0061	0.2581 ± 0.0035	0.4087 ± 0.0079	0.7736 ± 0.0039	0.0529 ± 0.0008
	TCMBN	0.4192 ±0.0064	0.1577 ± 0.0066	0.3687 ± 0.0065	0.8187 ± 0.0030	0.0530 ± 0.0005
	LANET(ours)	**0.6253 ± 0.0026**	**0.4916 ± 0.0082**	**0.6159 ± 0.0029**	**0.9445 ± 0.0008**	**0.0474 ± 0.0003**

Данный раздел посвящен основным экспериментальным результатам на разных наборах данных. Также были проведены различные исследования на устойчивость модели к разным изменениям компонент.

Метрики для сравнения нашего подхода LANET с устоявшимися моделями для задачи прогнозирования временных множеств представлены в таблице 6.1. LANET демонстрирует производительность top-1 на всех наборах данных, существенно превосходя своих конкурентов. Огромный разрыв в производительности наблюдается на DC, что может быть связано с огромным количеством доступных последовательностей для обучения в этом наборе данных или со специфическими структурами наборов. Ближайшим конкурентом LANET является модель TCMBN, которая также основана на архитектуре трансформера. Результаты показывают, что решающим моментом является обработка исторической информации на входе модели, а не ее последующая обработка. LANET успешно справляется с этой задачей и показывает совершенно другой уровень производительности. Интересно, что статистическая базовая линия GPTopFreq в некоторых случаях демонстрирует более высокое качество, чем модели глубоких нейронных сетей. Такое явление также упоминается в [39].

Для синтетического набора медицинских данных Synthea существует единый диапазон значений для всех моделей, что объясняется природой набора данных, так как он не такой сложный, как Mimic3. В Mimic3 разброс значений между LANET и другими моделями довольно большой. LANET лучше выявляет взаимозависимости и сложные закономерности в данных. Mimic3 имеет наименьшее количество событий, что влияет на сходимость моделей. В результате LANET имеет тенденцию быть быст-

рее остальных. Более того, если рассматривать набор данных DC, который имеет наибольшее количество событий и последовательностей, то наблюдается просадка качества других подходов из-за сложности структуры набора данных. Подводя итог, можно сказать, что конкурентоспособность есть со стороны модели TCMBN, которая также основана на архитектуре трансформера, но при этом правильная работа с метками побеждает.

6.1. Исследования и интерпретируемость

Мы исследуем зависимость производительности LANET от ее основных гиперпараметров.

6.1.1. Вклад временного векторного представления

Каждый набор в последовательности сопоставлен временной метке, которая участвует в получении временных представлений. Мы решили рассмотреть вклад временного компонента в производительность модели. Поэтому мы опускаем временную информацию из LANET, заменяя временные представления постоянным вектором. Такой вектор указывает на присутствие конкретной метки в истории пользователя, пренебрегая всеми временными зависимостями. Падение метрики в результате этой модификации приведено в Таблице 6.2. Однако даже после исключения временных представлений из LANET он по-прежнему демонстрирует повышенную производительность из-за эффективной обработки аналогичного частотного исторического представления.

Таблица 6.2: Вклад временной информации в производительность LANET.

Dataset	Model	F1	ROC-AUC
Synthea	No time	0.3890 ± 0.0162	0.8810 ± 0.0023
	LANET	0.4704 ± 0.0071	0.9026 ± 0.0018
Mimic III	No time	0.7644 ± 0.0023	0.9775 ± 0.0001
	LANET	0.8214 ± 0.0224	0.9852 ± 0.0023
DC	No time	0.4316 ± 0.0044	0.9906 ± 0.0000
	LANET	0.5498 ± 0.0137	0.9941 ± 0.0004
Instacart	No time	0.5277 ± 0.0032	0.9145 ± 0.0004
	LANET	0.6159 ± 0.0029	0.9445 ± 0.0008

6.1.2. Зависимость от размера векторного представления.

Существенной частью нашей модели является использование обучаемых представлений для управления временными наборами. По этой причине необходимо изучить влияние размерности эмбеддингов на метрики

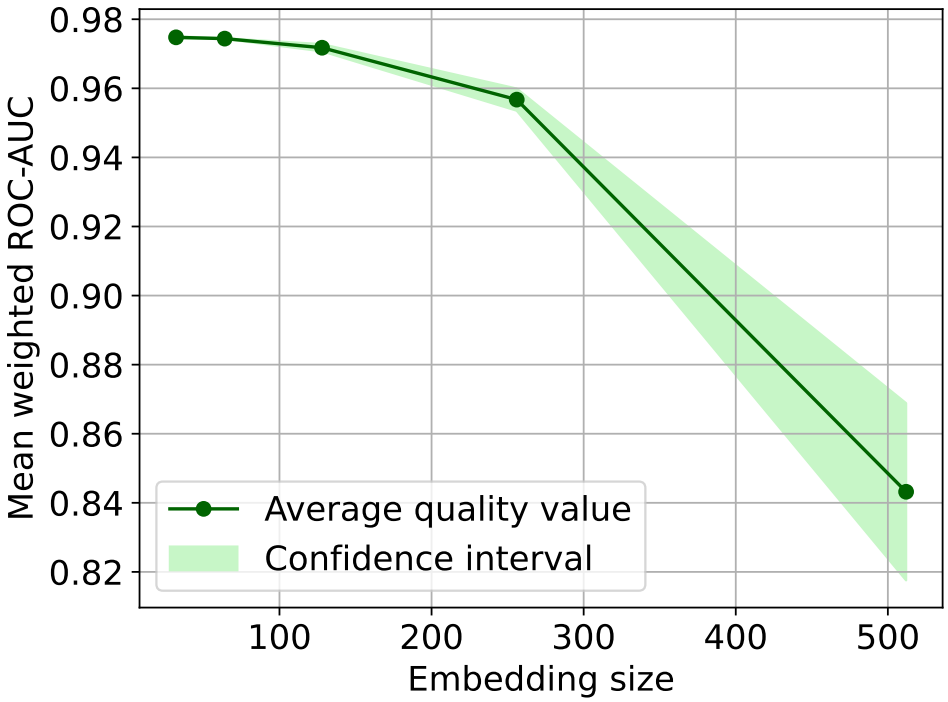

Рис. 6.1: Зависимость качества LANET от размера векторного представления.

LANET, поскольку этот параметр напрямую связан с емкостью модели. Размерность совместных представлений до передачи в блок энкодера равна $2D$. Эффект изменения значений D представлен на рисунке 6.1. Из него можно сделать вывод, что LANET испытывает трудности с эффективным изучением представлений высокой размерности.

Зависимость от количества голов Использование нескольких голов в слоях внимания позволяет модели учитывать несколько различных зависимостей, выделяя отдельную голову для понимания определенного шаблона. Рисунок 6.2 подтверждает, что более значительное количество принятых голов приводит к улучшению качества. Однако потребление ресурсов растет вместе с увеличением количества голов.

Зависимость от количества слоев в энкодере Гиперпараметр количества слоев энкодера отвечает за способность распознавать сложные взаимосвязи в данных. Рисунок 6.3 демонстрирует, что существует оптимальное количество слоев для решения рассматриваемой задачи. Дальнейшее увеличение количества слоев приводит к невозможности обучения эффективной модели.

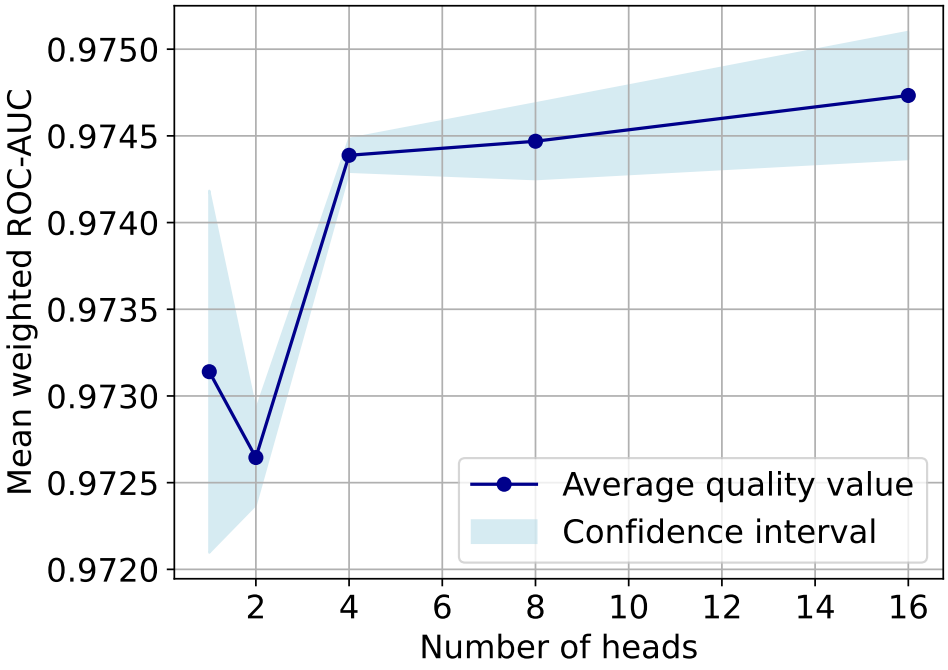

Рис. 6.2: Зависимость качества LANET от количества голов внимания.

6.1.3. Графическая интерпретация весов внимания

Важной частью полученной архитектуры является слой энкодера, который включает слой внимания. Внимание, в свою очередь, указывает на степень релевантности взаимосвязи между метками, что имеет значение для дальнейшего прогнозирования модели. Мы выбираем наиболее релевантные метки для выбора в Instacart, чтобы определить причинные объяснения прогнозов меток. Рисунок 6.5 слева показывает тепловую карту их взаимосвязей. Мы замечаем, что матрица внимания явно доминирует над метками, встречающимися в последовательности, над теми, которые в ней не находятся, что четко выражается через веса. Если посмотреть глубже, мы увидим, что мелкомасштабные вариации внимания описывают связь между определенными типами событий.

Кроме того, мы рассматриваем наиболее релевантные метки для выборки. Рисунок слева показывает тепловую карту их взаимосвязей. Для генерации причинных объяснений нам потребовалась графическая визуализация шкал внимания для отдельных меток. Это идея, лежащая в основе фреймворка CLEANN [59], который предлагает метод извлечения причинно-следственных связей в виде частичного предкового графа (PAG) [57]. Итак, для формирования графа мы рассмотрели одного из пользователей и соответствующую историческую информацию о метках. Используя предварительно обученную модель LANET, мы получили веса внимания, подаваемые в алгоритм CLEANN.

Левая визуализация графа на рисунке 6.4 имеет несколько типов связей:

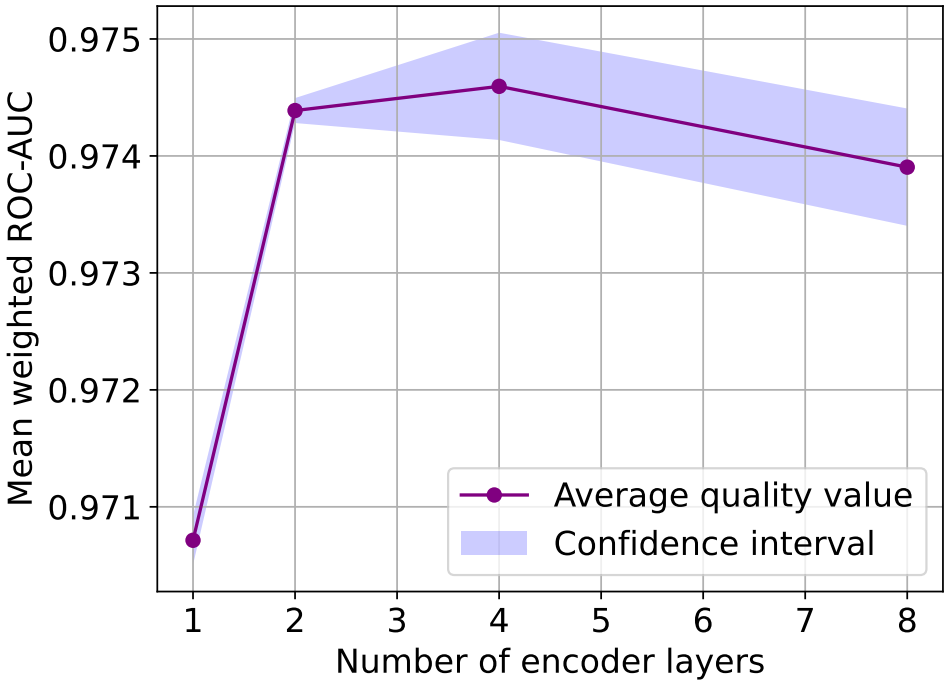

Рис. 6.3: Зависимость качества LANET от количества слоев энкодера.

- Красные линии указывают на близость меток внутри графа;
- Синие связи более сложные, это двунаправленное взаимодействие между метками в графе;
- Черный означает, что метка является родительской для последующей;
- Зеленый, наоборот, является потомком.

В первом случае между метками сложились сложные и запутанные связи. Например, если родительской является “консервированное мясо и морепродукты”, то вы сгенерируете “топпинги для салатной заправки”. Некоторые связи могут показаться нам нелогичными, но эта история индивидуальна для каждого пользователя при покупке товаров в магазине.

Более того, чтобы выяснить и идентифицировать связи, мы решили удалить метку с наибольшим общим весом в матрице внимания и посмотреть на перераспределение весов в этом случае (рисунок 6.6 справа). Модель переключила внимание на множество других меток. PAG демонстрирует измененную картину, где исчезли все синие и черные ребра графика, что соответствует более сложной и ориентированной связи, чем простая “соседская”. Корреляция между метками стала ниже. Более того, “консервированное мясо и морепродукты” изменила свое поведение. Она стала дочерней и больше ни с кем не связана, что влияет на предсказательную способность этой метки на следующий временной шаг.

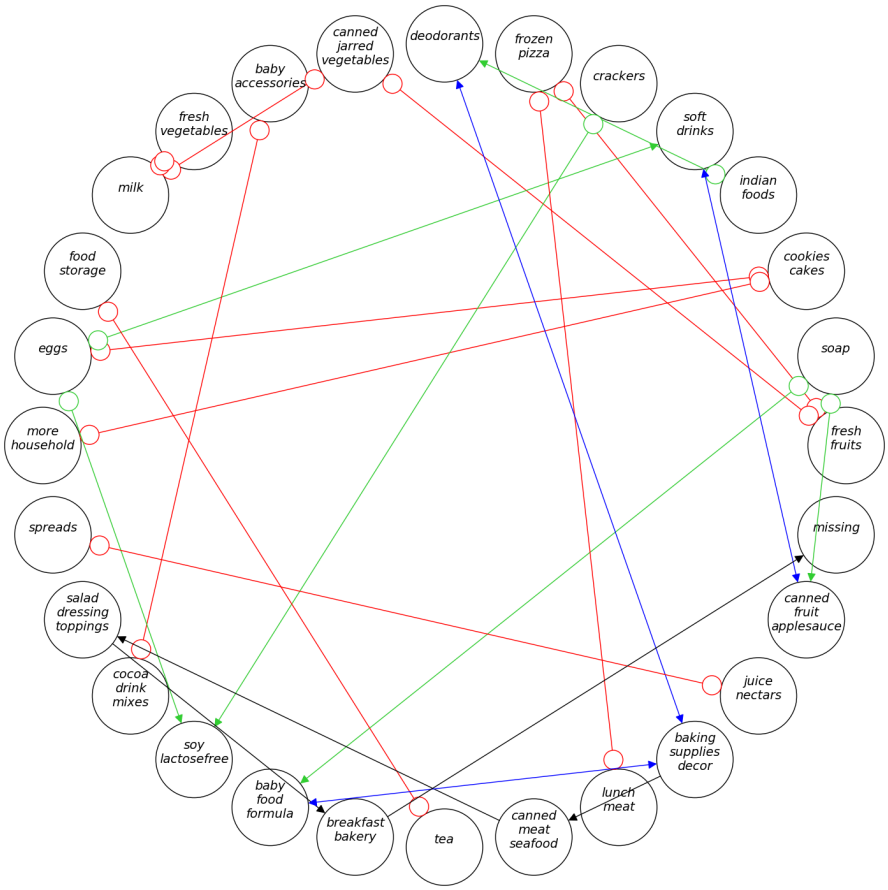

Рис. 6.4: График взаимосвязи между подмножеством меток и их вербальной интерпретацией (исходные данные)

Это исследование показывает, что наилучшие прогностические возможности LANET в основном зависят от способности модели обнаруживать взаимосвязи между метками, а не от построения работы со временем и порядком, в котором размещены корзины.

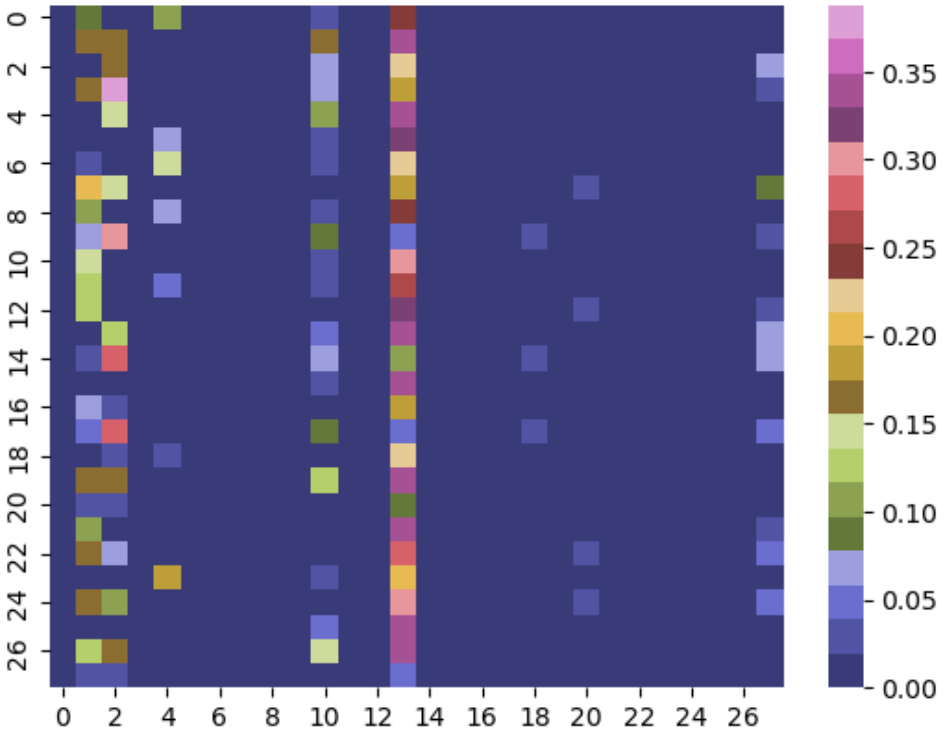

Рис. 6.5: Тепловая карта взаимосвязей всех меток набора данных Instacart (исходные данные)

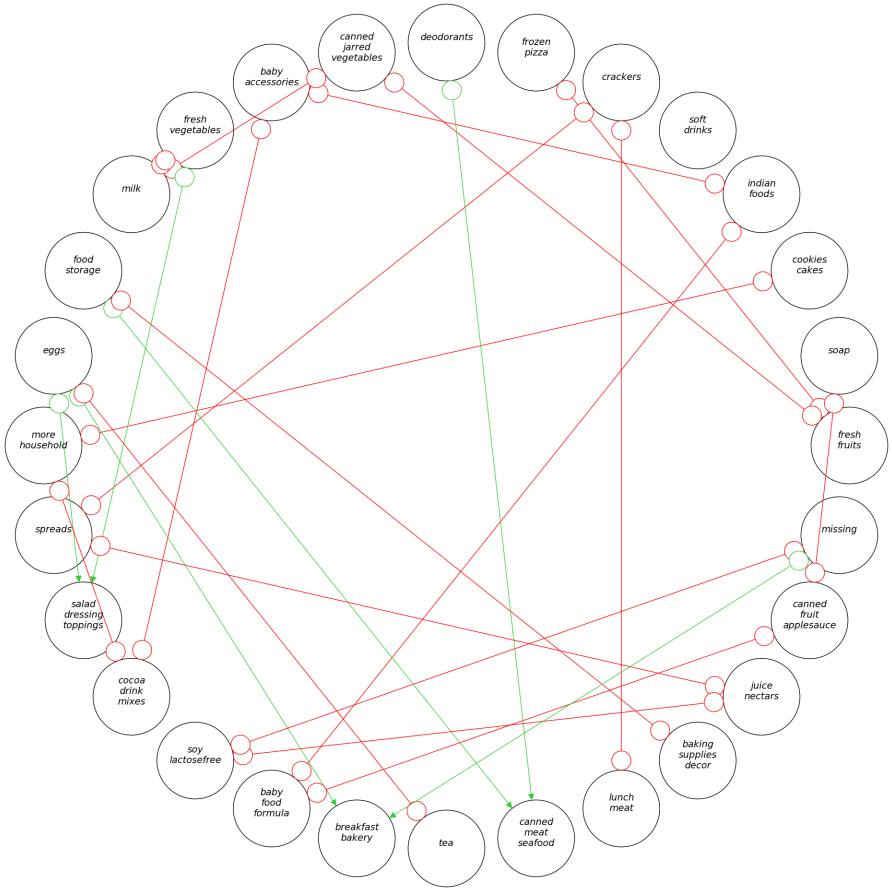

Рис. 6.6: Модифицированный график после удаления метки с наибольшим весом внимания

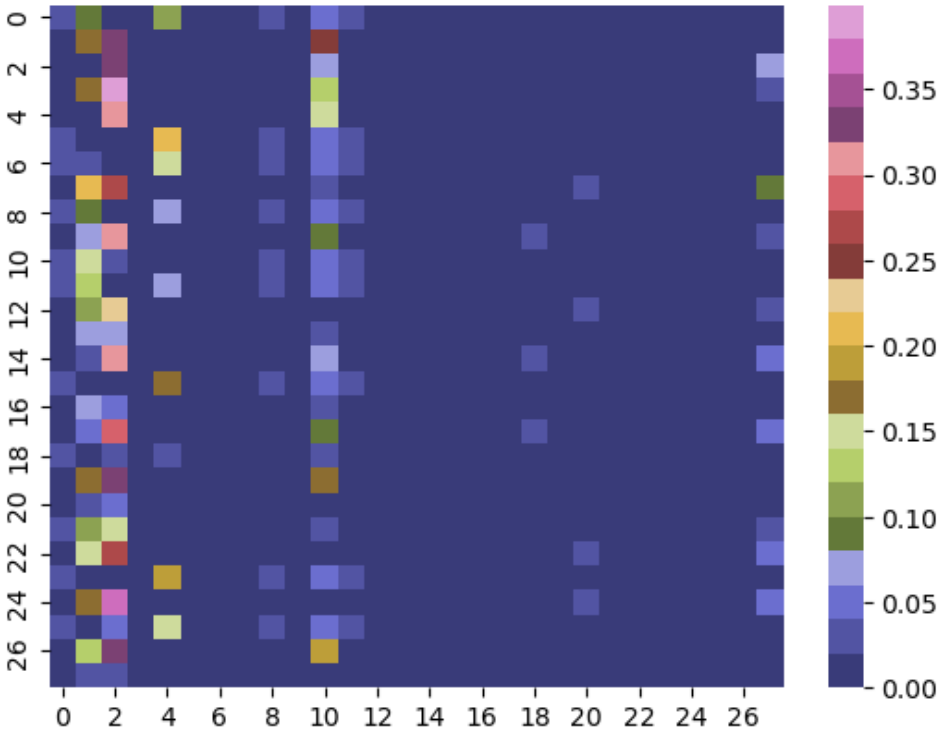

Рис. 6.7: Тепловая карта после удаления метки с максимальным весом внимания

Глава 7
Заключение

В данной работе была рассмотрена задача прогнозирования временных наборов меток, которая имеет широкое применение в таких областях, как электронная коммерция, финансы и медицина. Основная сложность задачи заключается в необходимости одновременного учета временных зависимостей между событиями и взаимосвязей между метками внутри наборов. Для её решения была предложена модель LANET, основанная на трансформерах и механизме внимания, которая демонстрирует значительное улучшение качества прогнозирования по сравнению с существующими аналогами.

Основные результаты работы:

- Предложена модель LANET — архитектура, которая эффективно агрегирует историческую информацию о временных наборах меток в специальные представления, учитывающие как временные зависимости, так и взаимосвязи между метками.
- Проведено сравнение LANET со современными методами (SFCNTSP, DNNTSP, GPTopFreq, TCMBN) на четырех наборах данных (Synthea, Mimic III, DC, Instacart). LANET показала наилучшие результаты, улучшив метрику Weighted F1 до 65% по сравнению с ближайшим конкурентом.
- Изучено влияние компонентов модели на её производительность, включая анализ вклада временной информации, размерности эмбеддингов, количества голов внимания и слоев трансформера. Показано, что LANET в большей степени ориентирована на выявление взаимосвязей между метками, чем на временные зависимости.
- Проведена интерпретация механизма внимания с помощью визуализации графов взаимодействий меток, что позволило выявить причинно-следственные связи между ними.
- Определены ограничения модели, такие как зависимость её эффективности от размера словаря меток, и намечены направления для дальнейших исследований, включая адаптацию к задачам с большим или малым количеством меток.

7.1. Перспективы дальнейших исследований

Адаптация LANET для задач с очень большим (тысячи) или малым (5–20) количеством меток. Изучение влияния агрегации событий

в временные наборы на качество прогнозирования.Применение подхода в парадигме self-supervised learning для получения универсальных представлений данных.

Таким образом, LANET представляет собой эффективное решение для прогнозирования временных наборов меток, сочетающее в себе высокую точность и интерпретируемость. Результаты работы открывают новые возможности для применения модели в различных областях, включая рекомендательные системы, анализ транзакций и медицинскую диагностику.

Список литературы

[1] https://pytorch.org/docs/stable/generated/torch.nn.TransformerEncoderLayer.html.

[2] *Amazon Product Reviews*. https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews. 2018.

[3] Mozhdeh Ariannezhad и др. «A personalized neighborhood-based model for within-basket recommendation in grocery shopping». В: *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 2023, с. 87—95.

[4] Dmitrii Babaev и др. «CoLES: Contrastive Learning for Event Sequences with Self-Supervision». В: *Proceedings of the 2022 International Conference on Management of Data*. 2022, с. 1190—1199.

[5] Alexandra Bazarova и др. «Universal representations for financial transactional data: embracing local, global, and external contexts». В: *arXiv preprint arXiv:2404.02047* (2024).

[6] Patrik Berger и Michal Kompan. «User modeling for churn prediction in E-commerce». В: *IEEE Intelligent Systems* 34.2 (2019), с. 44—52.

[7] Tianqi Chen и Carlos Guestrin. «Xgboost: A scalable tree boosting system». В: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, с. 785—794.

[8] Wen-Hao Chiang, Xueying Liu и George Mohler. «Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates». В: *International journal of forecasting* 38.2 (2022), с. 505—520.

[9] Kyunghyun Cho и др. «Learning phrase representations using RNN encoder-decoder for statistical machine translation». В: *arXiv preprint arXiv:1406.1078* (2014).

[10] Jae Young Choi и Bumshik Lee. «Combining LSTM network ensemble via adaptive weighting for improved time series forecasting». В: *Mathematical Problems in Engineering* 2018 (2018).

[11] Tri Dao и др. «Hungry Hungry Hippos: Towards Language Modeling with State Space Models». В: *ICLR*. 2023.

[12] Angus Dempster, François Petitjean и Geoffrey Webb. «ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels». В: *Data Mining and Knowledge Discovery* 34 (сент. 2020). DOI: 10.1007/s10618-020-00701-z.

[13] Jingcheng Du и др. «ML-Net: multi-label classification of biomedical texts with deep neural networks». В: *Journal of the American Medical Informatics Association* 26.11 (2019), с. 1279—1285.

[14] Nan Du и др. «Recurrent Marked Temporal Point Processes: Embedding Event History to Vector». В: KDD. San Francisco, California, USA: Association for Computing Machinery, 2016, с. 1555—1564. ISBN: 9781450342322. DOI: 10.1145/2939672.2939875.

[15] *Dunnhumby-Carbo*. https://www.dunnhumby.com/source-files/. 2020.

[16] Thibaut Durand, Nazanin Mehrasa и Greg Mori. «Learning a deep convnet for multi-label classification with partial labels». В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, с. 647—657.

[17] Ivan Fursov и др. «Adversarial attacks on deep models for financial transaction records». В: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, с. 2868—2878.

[18] Ivan Fursov и др. «Gradient-based adversarial attacks on categorical sequence models via traversing an embedded world». В: *Analysis of Images, Social Networks and Texts: AIST*. Springer. 2021, с. 356—368.

[19] Albert Gu, Karan Goel и Christopher Re. «Efficiently Modeling Long Sequences with Structured State Spaces». В: *ICLR*. 2022.

[20] Jun-Yi Hang и Min-Ling Zhang. «Collaborative Learning of Label Semantics and Deep Label-Specific Features for Multi-Label Classification». В: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[21] Thomas Hartvigsen и др. «Recurrent halting chain for early multi-label classification». В: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, с. 1382—1392.

[22] Alan G Hawkes. «Spectra of some self-exciting and mutually exciting point processes». В: *Biometrika* 58.1 (1971), с. 83—90.

[23] Kaiming He и др. «Masked autoencoders are scalable vision learners». В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, с. 16000—16009.

[24] Sepp Hochreiter и Jürgen Schmidhuber. «Long short-term memory». В: *Neural computation* 9.8 (1997), с. 1735—1780.

[25] *Instacart Market Basket Analysis*. https://www.kaggle.com/c/instacart-market-basket-analysis/data. 2017.

[26] Himanshu Jain, Yashoteja Prabhu и Manik Varma. «Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications». В: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, с. 935—944.

[27] Alistair EW Johnson и др. «MIMIC-III, a freely accessible critical care database». В: *Scientific data* 3.1 (2016), с. 1—9.

[28] Wang-Cheng Kang и Julian McAuley. «Self-attentive sequential recommendatio В: *2018 IEEE international conference on data mining (ICDM)*. IEEE. 2018, с. 197—206.

[29] Jacob Devlin Ming-Wei Chang Kenton и Lee Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». В: *Proceedings of NAACL-HLT*. 2019, с. 4171—4186.

[30] Sanghyeon Kim и Eunbyung Park. «Smpconv: Self-moving point representations for continuous convolution». В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, с. 10289—10299.

[31] Diederik P. Kingma и Jimmy Ba. «Adam: A Method for Stochastic Optimization». В: *3rd ICLR, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Под ред. Yoshua Bengio и Yann LeCun. 2015.

[32] John Frank Charles Kingman. *Poisson processes*. Т. 3. Clarendon Press, 1992.

[33] Philipp Koch и др. «Recurrent neural networks with weighting loss for early prediction of hand movements». В: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE. 2018, с. 1152—1156.

[34] Kamesh Korangi, Christophe Mues и Cristián Bravo. «A transformer-based model for default prediction in mid-cap corporate markets». В: *arXiv preprint arXiv:2111.09902* (2021).

[35] Jack Lanchantin и др. «General Multi-Label Image Classification With Transformers». В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Июнь 2021, с. 16478—16488.

[36] Jerald Franklin Lawless. «Regression methods for Poisson process data». В: *Journal of the American Statistical Association* 82.399 (1987), с. 808—815.

[37] Jure Leskovec и Andrej Krevl. *SNAP Datasets: Stanford large network dataset collection*. 2014.

[38] Jiacheng Li, Yujie Wang и Julian McAuley. «Time interval aware self-attention for sequential recommendation». В: *Proceedings of the 13th international conference on web search and data mining*. 2020, с. 322—330.

[39] Ming Li и др. «A next basket recommendation reality check». В: *ACM Transactions on Information Systems* 41.4 (2023), с. 1—29.

[40] Yuncheng Li, Yale Song и Jiebo Luo. «Improving Pairwise Ranking for Multi-Label Image Classification». В: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Июль 2017.

[41] Minghao Liu и др. «Gated transformer networks for multivariate time series classification». В: *arXiv preprint arXiv:2103.14438* (2021).

[42] Weiwei Liu и др. «The emerging trends of multi-label learning». В: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[43] Michal Lukasik и др. «Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter». В: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016, с. 393—398.

[44] Dixin Luo и др. «You are what you watch and when you watch: Inferring household structures from IPTV viewing data». В: *IEEE Transactions on Broadcasting* 60.1 (2014), с. 61—72.

[45] Anton Lysenko, Egor Shikov и Klavdiya Bochenina. «Temporal point processes for purchase categories forecasting». В: *Procedia Computer Science* 156 (2019), с. 255—263.

[46] Alexander Marusov и Alexey Zaytsev. «Non-contrastive representation learning for intervals from well logs». В: *IEEE Geoscience and Remote Sensing Letters* (2023).

[47] Hongyuan Mei и Jason Eisner. «The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process». В: NeurIPS. Long Beach, California, USA: Curran Associates Inc., 2017, с. 6757—6767. ISBN: 9781510860964.

[48] Hongyuan Mei и Jason M Eisner. «The neural Hawkes process: A neurally self-modulating multivariate point process». В: *NeurIPS* 30 (2017).

[49] Hongyuan Mei, Chenghao Yang и Jason Eisner. «Transformer embeddings of irregularly spaced events and their participants». В: *International Conference on Learning Representations*. 2021.

[50] Aditya K Menon и др. «Multilabel reductions: what is my loss optimising?» В: *Advances in Neural Information Processing Systems* 32 (2019).

[51] *MIMIC-III Clinical Database*. https://physionet.org/content/mimiciii/1.4/. 2016.

[52] Aäron van den Oord и др. «WaveNet: A Generative Model for Raw Audio». В: *9th ISCA Speech Synthesis Workshop*. 2016, с. 125—125.

[53] *Open Synthetic Patient Data*. https://github.com/lhs-open/synthetic-data. 2022.

[54] Ankit Pal, Muru Selvakumar и Malaikannan Sankarasubbu. «Multi-label text classification using attention-based graph neural network». В: *arXiv preprint arXiv:2003.11644* (2020).

[55] Liudmila Prokhorenkova и др. «CatBoost: unbiased boosting with categorical features». В: *NeurIPS*. Под ред. S. Bengio и др. Т. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

[56] Massimo Quadrana, Paolo Cremonesi и Dietmar Jannach. «Sequence-aware recommender systems». В: *ACM Computing Surveys (CSUR)* 51.4 (2018), с. 1—36.

[57] Thomas Richardson и Peter Spirtes. «Ancestral graph Markov models». В: *The Annals of Statistics* 30.4 (2002), с. 962—1030.

[58] Marian-Andrei Rizoiu и др. «Hawkes processes for events in social media». В: *Frontiers of multimedia research*. 2017, с. 191—218.

[59] Raanan Y Rohekar, Yaniv Gurwicz и Shami Nisimov. «Causal Interpretation of Self-Attention in Pre-Trained Transformers». В: *Advances in Neural Information Processing Systems* 36 (2024).

[60] David W Romero и др. «CKConv: Continuous Kernel Convolution For Sequential Data». В: *ICLR*. 2021.

[61] David W. Romero и др. «CKConv: Continuous Kernel Convolution For Sequential Data». В: *The Tenth ICLR, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[62] David W. Romero и др. *FlexConv: Continuous Kernel Convolutions with Differentiable Kernel Sizes*. 2022. arXiv: 2110.08059 [cs.CV].

[63] Avirup Saha и др. «Learning Network Traffic Dynamics Using Temporal Point Process». В: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2019, с. 1927—1935. DOI: 10.1109/INFOCOM.2019.8737622.

[64] Sofia Serrano и Noah A Smith. «Is Attention Interpretable?» В: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, с. 2931—2951.

[65] Oleksandr Shchur и др. «Neural temporal point processes: A review». В: *arXiv preprint arXiv:2104.03528* (2021).

[66] Hui Shi и др. «Continuous CNN for nonuniform time series». В: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, с. 3550—3554.

[67] Xiao Shou и др. «Concurrent Multi-Label Prediction in Event Streams». В: *AAAI Conference on Artificial Intelligence*. 2023.

[68] Vincent Sitzmann и др. «Implicit Neural Representations with Periodic Activation Functions». В: *NeurIPS*. Под ред. H. Larochelle и др. Т. 33. Curran Associates, Inc., 2020, с. 7462—7473.

[69] Tengshuo Song и др. «HGAT-BR: Hyperedge-based graph attention network for basket recommendation». В: *Applied Intelligence* 53.2 (2023), с. 1435—1451.

[70] Fei Sun и др. «BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer». В: *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019, с. 1441—1450.

[71] Jiaxi Tang и Ke Wang. «Personalized top-n sequential recommendation via convolutional sequence embedding». В: *Proceedings of the eleventh ACM international conference on web search and data mining*. 2018, с. 565—573.

[72] Adane Nega Tarekegn, Mario Giacobini и Krzysztof Michalak. «A review of methods for imbalanced multi-label classification». В: *Pattern Recognition* 118 (2021), с. 107965.

[73] Yi Tay и др. «Efficient transformers: A survey». В: *ACM Computing Surveys* 55.6 (2022), с. 1—28.

[74] *The event logs involving error reporting and failure tickets*. https://github.com/woshiyyya/ERPP-RMTPP. 2019.

[75] Oskar Triebe и др. «NeuralProphet: Explainable Forecasting at Scale». В: *arXiv preprint arXiv:2111.15397* (2021).

[76] Che-Ping Tsai и Hung-Yi Lee. «Order-free learning alleviating exposure bias in multi-label classification». В: *Proceedings of the AAAI Conference on Artificial Intelligence*. Т. 34. 04. 2020, с. 6038—6045.

[77] Shashanka Ubaru и др. «Multilabel Classification by Hierarchical Partitioning and Data-dependent Grouping». В: *Advances in Neural Information Processing Systems* 33 (2020), с. 22542—22553.

[78] Luuk Van Maasakkers, Dennis Fok и Bas Donkers. «Next-basket prediction in a high-dimensional setting using gated recurrent units». В: *Expert Systems with Applications* 212 (2023), с. 118795.

[79] Ashish Vaswani и др. «Attention is all you need». В: *Advances in neural information processing systems* 30 (2017).

[80] Shoujin Wang, Liang Hu и Longbing Cao. «Perceiving the next choice with comprehensive transaction embeddings for online recommendation». В: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 17*. Springer. 2017, с. 285—302.

[81] Shoujin Wang и др. «Sequential recommender systems: challenges, progress and prospects». В: *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*. International Joint Conferences on Artificial Intelligence. 2019, с. 6332—6338.

[82] Wei Wei и др. «Effective detection of sophisticated online banking fraud on extremely imbalanced data». В: *World Wide Web* 16 (2013), с. 449—475.

[83] Chao-Yuan Wu и др. «Recurrent recommender networks». В: *Proceedings of the tenth ACM international conference on web search and data mining*. 2017, с. 495—503.

[84] Haixu Wu и др. «Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting». В: *NeurIPS*. Под ред. M. Ranzato и др. Т. 34. Curran Associates, Inc., 2021, с. 22419—22430.

[85] Xi-Zhu Wu и Zhi-Hua Zhou. «A unified view of multi-label performance measures». В: *International Conference on Machine Learning*. PMLR. 2017, с. 3780—3788.

[86] Lin Xiao и др. «Label-specific document representation for multi-label text classification». В: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 2019, с. 466—475.

[87] Hongteng Xu и Hongyuan Zha. «A Dirichlet mixture model of Hawkes processes for event sequence clustering». В: *NeurIPS* 30 (2017).

[88] Junchi Yan. «Recent advance in temporal point process: from machine learning perspective». В: *SJTU Technical Report* (2019).

[89] V Ramanjaneyulu Yannam и др. «Hybrid approach for next basket recommendation system». В: *International Journal of Information Technology* 15.3 (2023), с. 1733—1740.

[90] Vacit Oguz Yazici и др. «Orderless recurrent models for multi-label classification». В: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, с. 13440—13449.

[91] Chih-Kuan Yeh и др. «Learning deep latent space for multi-label classification». В: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[92] Haochao Ying и др. «Sequential recommender system based on hierarchical attention network». В: *IJCAI International Joint Conference on Artificial Intelligence*. 2018.

[93] Le Yu и др. «Continuous-time user preference modelling for temporal sets prediction». В: *IEEE Transactions on Knowledge and Data Engineering* (2023).

[94] Le Yu и др. «Predicting temporal sets with simplified fully connected networks». В: *Proceedings of the AAAI Conference on Artificial Intelligence*. Т. 37. 4. 2023, с. 4835—4844.

[95] Zhihan Yue и др. «Ts2vec: Towards universal representation of time series». В: *Proceedings of the AAAI Conference on Artificial Intelligence*. Т. 36. 8. 2022, с. 8980—8987.

[96] Ailing Zeng и др. «Are Transformers Effective for Time Series Forecasting?» В: 2023.

[97] Fan Zhang и др. «Conv-based Temporal Sets Prediction for Next-basket Recommendation». В: *2023 International Conference on Frontiers of Robotics and Software Engineering (FRSE)*. IEEE. 2023, с. 419—425.

[98] Min-Ling Zhang и Zhi-Hua Zhou. «A review on multi-label learning algorithms». В: *IEEE transactions on knowledge and data engineering* 26.8 (2013), с. 1819—1837.

[99] Qiang Zhang и др. «Self-attentive Hawkes process». В: *ICML*. PMLR. 2020, с. 11183—11193.

[100] Shuai Zhang и др. «Next item recommendation with self-attentive metric learning». В: *Thirty-Third AAAI Conference on Artificial Intelligence*. Т. 9. 2019.

[101] Wenyu Zhang и др. «Multi-label prediction in time series data using deep neural networks». В: *arXiv preprint arXiv:2001.10098* (2020).

[102] Qingyuan Zhao и др. «Seismic: A self-exciting point process model for predicting tweet popularity». В: *ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, с. 1513—1522.

[103] Ke Zhou, Hongyuan Zha и Le Song. «Learning triggering kernels for multi-dimensional hawkes processes». В: *ICML*. PMLR. 2013, с. 1301—1309.

[104] Vladislav Zhuzhel и др. «COHORTNEY: Non-Parametric Clustering of Event Sequences». В: *arXiv preprint arXiv:2104.01440* (2021).

[105] Vladislav Zhuzhel и др. *Continuous-time convolutions model of event sequences*. 2023. arXiv: 2302.06247 [cs.LG].

[106] Simiao Zuo и др. «Transformer Hawkes process». В: *ICML*. PMLR. 2020, с. 11692—11702.