

[language=R] «echo=FALSE,results=tex»= cat("dataset ") cat("textbfdataset ") it should be bold @

---

# Label Attention Network для последовательной классификации по нескольким меткам.

---

Боева Галина  
Антиплагиат  
Сколтех  
boeva.gl@phystech.edu

Консультант: к.ф.-м.н. Грабовой Андрей  
Антиплагиат  
grabovoy.av@phystech.edu

Эксперт: к.ф.-м.н. Зайцев Алексей  
Сколтех  
a.zaytsev@skoltech.ru

8 апреля 2024 г.

## Аннотация

Рассматривается задача прогнозирования временных наборов для последовательных данных. Современные подходы фокусируются на архитектуре преобразования последовательных данных, используя собственного внимания (“self-attention”) к элементам в последовательности. В этом случае учитываются временные взаимодействия событий, но теряем информацию о взаимозависимостях меток. Мотивированные этим недостатком, предлагается использовать механизм собственного внимания (“self-attention”) к меткам, предшествующим прогнозируемому шагу. Поскольку рассматриваемый подход представляет собой сеть внимания к меткам — LANET. В данном исследовании обосновывается вывод причинно-следственной связи внимания, которое указывает на важность меток и их взаимозависимости.

Ключевые слова временные ряды · взаимосвязь меток

## 1 Введение

Классификация с несколькими метками является более естественной, чем бинарная или многоклассовая классификация, поскольку все, что окружает нас в реальном мире, обычно описывается несколькими метками [1]. Та же логика может быть перенесена на последовательность событий с отметками времени. События в последовательности, как правило, характеризуются несколькими категориальными значениями вместо одной. Существует множество подходов к классификации с несколькими метками в компьютерном зрении [2], обработке естественного языка [3] или классической структуре табличных данных [4]. Однако постановке задачи с несколькими метками для последовательностей событий, как правило, уделяется меньше внимания. Итак, основной целью является противостоять такому недостатку внимания и решить проблему предсказания набора меток для последовательных данных с временными метками. Важно отметить, что модель должна предсказывать набор меток, соответствующих следу-

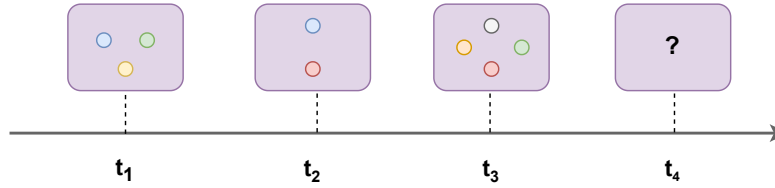


Рис. 1: На рисунке показано визуальное представление постановки задачи. Наша модель должна предсказать метки для момента времени  $t_4$ , учитывая историю предыдущих наборов меток. Требуется предсказать несколько меток, так что это определение задачи классификации с несколькими метками.

ющему шагу, принимая во внимание содержимое предыдущих групп меток для последовательности событий, связанных с объектом (Рисунок 1).

Взаимодействие между состояниями объекта в разные временные метки имеет важное значение для решения задач с последовательными данными [5]. Следовательно, выразительные и мощные модели должны быть способны изучать такие взаимодействия. Несколько архитектур нейронных сетей, таких как трансформеры [6] или рекуррентные нейронные сети [7], способны делать это. Например, трансформер напрямую определяет механизм внимания, который измеряет, как связаны различные временные метки в последовательности. Однако применение современных методов глубокого обучения ограничено [8], и они в первую очередь сосредоточены на прогнозировании меток для последовательности в целом.

### 1.1 Основные подходы для задачи классификации с несколькими метками.

Постановка задачи классификации с несколькими метками возникает во многих различных областях, например, при категоризации текста или тегировании изображений, и все они влекут за собой свои собственные особенности и проблемы. В обзоре [9] исследуются основы обучения с использованием нескольких меток, обсуждаются хорошо зарекомендовавшие себя методы, а также самые последние подходы. Возникающие тенденции рассматриваются в более свежем обзоре [1].

В работе [10] рассматривается та же постановка задачи классификации с несколькими метками в потоке событий, что и у нас. Модель авторов нацелена на фиксацию временных и вероятностных зависимостей между типами параллельных событий путем кодирования исторической информации с помощью энкодера, а затем использования условной смеси экспертов Бернулли. В этой статье [11] обсуждается постановка задачи прогнозирования временных наборов для пользователей, она предлагает систему непрерывного обучения, которая позволяет явно фиксировать изменяющиеся пользовательские предпочтения, поддерживая банк памяти, который мог бы хранить состояния всех пользователей и элементов. В этой парадигме авторы строят неубывающую универсальную последовательность, содержащую все пользовательские взаимодействия, а затем в хронологическом порядке извлекают уроки из каждого взаимодействия. Для исследования взаимосвязи между продуктами в корзине был предложен ConvTSP [12], который объединяет динамические интересы пользователей и статистические интересы в единое векторное представление.

### 1.2 Рекомендательные системы.

В этом разделе мы представим статьи, связанные с проблемой рекомендации следующей корзины. Эта формулировка похожа на нашу, поэтому мы также рассмотрели многие подходы и идеи при анализе нашей области исследований. Авторы в [13] предложили персонализированную модель, которая фиксирует краткосрочные зависимости внутри временного набора продуктов, а также долгосрочную, основанную на исторической информации о пользователях. Также в [14] для соединения локальной и глобальной пользовательской информации предлагается гибридный метод, основанный на автоэнкодере [15] для извлечения контекста и рекуррентные нейронные сети для понимания динамики изменения интересов. Чтобы преодолеть подобные проблемы, для предсказания следующей рекомендации создается сеть внимания на основе графов, использующая hyper-edge подход [16]. При такой постановке задачи возникает сложность работы со словарем товарных категорий, поскольку они насчитывают тысячи значений, в исследовании [17] используется GRU для прогнозирования следующей корзины, которая легко масштабируется до большого ассортимента.

Вклад. Разработана архитектура на основе трансформера на основе собственного внимания между метками для работы над задачей классификации последовательностей событий по нескольким меткам. Основной вклад заключается в следующем:

- Была введена архитектура LANET для прогнозирования набора меток для текущего события, используя информацию из предыдущих событий. Особенностью архитектуры является вычисление собственного внимания между представлениями меток.
- LANET превосходит модели на основе трансформера, которые фокусируются на вычислении собственного внимания между временными метками. Оцениваются все показатели в различных наборах данных (будет дополнение).

## 2 Постановка задачи

Задача классификации с несколькими метками это задача изучения функции  $f$ , которая сопоставляет входные данные с подмножествами набора меток  $\mathcal{L} = \{1, 2, \dots, L\}$ . Рассмотрим набор из  $N$  последовательностей  $D = (\mathbf{X}_n, \mathbf{Y}_n)_{n=1}^N$ . Предполагается, что пары  $(\mathbf{X}_n, \mathbf{Y}_n)$  являются случайными величинами, которые подчиняются неизвестному распределению  $P(\mathbf{X}, \mathbf{Y})$ . Пусть  $T_n = |\mathbf{Y}_n|$  обозначает размер набора меток, связанного с  $\mathbf{X}_n$ . В нашей постановке есть ограничение на размер прошлого, доступного модели. Обозначим  $\tau$  означает количество событий, предшествующих рассматриваемому событию, тогда  $D_t^u = (\mathbf{X}_j^u, \mathbf{Y}_j^u)_{j=t-\tau}^{t-1}$ , следующих одному и тому же неизвестному распределению  $P(\mathbf{X}, \mathbf{Y})$ . Введем функцию  $f(\cdot) \in [0, 1]^K$ , которая принимает историческую информацию о событиях в качестве входных данных и выводит вектор оценок для каждой из меток  $K$ . Эти оценки представляют собой вероятности присутствия метки в следующем наборе, связанных с событием. Итак, более формально  $f(\cdot)$  имеет вид:

$$f : D_t^u \rightarrow [0, 1]^K$$

для предсказания  $Y_t$ . Цель изучения  $f$  - минимизировать ожидаемую потерю на подвыборке в соответствии с базовым распределением  $P$ :

$$\min_f \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim P} [Loss(\mathbf{Y}, f(\mathbf{X}))],$$

где  $Loss$  — это логарифмическая функция потерь.

Чтобы завершить прогноз, нам нужна отдельная модель принятия решений о метках  $g(f(\cdot))$ , которая преобразует доверительные баллы в метки. Например, можно сравнить оценку для  $k$ -й метки с выбранным пороговым значением  $\beta_k$ : если  $f_k(\cdot) > \beta_k$ ,  $k = 1, \dots, K$ , то модель предсказывает, что  $k$ -я метка присутствует. Таким образом, модель  $g$  создает набор меток  $\hat{Y}_t \subseteq \mathcal{Y}$  на основе входных оценок достоверности.

## 3 Архитектура LANET

Большинство моделей, связанных с трансформаторами, используемых для последовательного предсказания с несколькими метками, используют вычисление собственного внимания между последовательными представлениями входных временных меток. Вместо этого LANET использует собственное внимание между представлениями меток. Итак, у него есть входные данные, состоящие из  $K$  векторов. Ниже описывается, как агрегировать последовательность векторов размером от  $\tau$  до  $K$  с помощью Слоя векторных представлений. Затем определяется Слой собственного внимания. Чтобы получить предсказания, применяется Слой предсказания.

Слой векторных представлений. Используется следующий подход для использования различных частей входных данных для последовательностей событий с несколькими метками:

- Векторное представление идентификаторов: Для идентификаторов изучается матрица эмбедингов;
- Векторное представление времени: Для каждой временной метки мы знаем значение  $dt$ , которое равно разнице в днях между рассматриваемой и предыдущей временной меткой. Проводим обучение представлений для каждого наблюдаемого значения  $dt$ . Также учитывается порядок событий, поэтому мы смотрим на представления для позиций:  $1, \dots, \tau$ , чтобы добавить их к  $dt$  представлению соответствующей временной метки;

Dataset	# events	Median set size	Max set size	# unique labels	Diff
Sales	47 217	16	48	84	0.0632
Demand	5 912	13	24	33	0.0957

Таблица 1: Характеристики наборов данных, используемых в задачах последовательной классификации с несколькими метками.

- Представление суммы: Все суммы преобразуются в ячейки, разбивая непрерывные суммы сумм на интервалы. Каждому интервалу присваивается уникальный номер. Затем для каждого уникального номера создается представление.
- Transformer label encoder: Для построения входных данных LANET используются данные, связанные с определенным идентификатором для временных меток  $\tau$ . Далее идет объединение представлений меток, которые встречаются во время временных меток  $\tau$ , с соответствующими векторами времени и суммы. Если метка не принадлежит истории последних временных меток  $\tau$ , к ней добавляются векторы нулей в виде представлений времени и суммы. Если конкретная метка встречается несколько раз на предыдущих шагах  $\tau$ , то создаются временные представления для каждого отдельного вхождения, а затем суммируются их, чтобы получить окончательное представление времени для этой метки. То же самое происходит при построении вектора суммы в ситуации повторения метки.

Итак, в результате слоя вложения у нас есть  $K + 1$  векторов вложения. Первый вектор  $\mathbf{e}_0$  соответствует вложениям общих (ID) признаков для последовательности. Все остальные векторы  $\mathbf{e}_k, k = 1, \dots, K$  являются объединением вложения метки с соответствующими вложениями времени и суммы. Оказывается, что векторы для исторически не задействованных меток — это просто вложения меток, поскольку мы суммируем их с нулевыми векторами времени и количества просмотров. При обучении всех весов встраивания они инициализируются из нормального распределения  $\mathcal{N}(0, 1)$  и затем оптимизируются.

Собственное внимание. После получения представлений из наших данных давайте перейдем к компоненте нашей архитектуры. Пусть  $E = \{\mathbf{e}_0, \dots, \mathbf{e}_K\}$ ,  $\mathbf{e}_i \in \mathbb{R}^d$  - последовательность входных представлений в энкодер, где  $\mathbf{e}_0$  соответствует представлению  $\mathbf{z}$  и все остальные  $\mathbf{e}_i$  соответствуют представлениям, фиксирующим историческую информацию с точки зрения метки. В архитектуре Transformer влияние представление  $\mathbf{e}_j$  на представление  $\mathbf{e}_i$  достигается с помощью собственного внимания. Вес внимания  $\alpha_{ij}$  и обновленный вектор  $\mathbf{e}'_i$  рассчитываются как:

$$\mathbf{e}'_i = \sum_{j=0}^K \alpha_{ij} (W^v \mathbf{e}_j); \quad \alpha_{ij} = \text{softmax} \left( (W^q \mathbf{e}_i)^T (W^k \mathbf{e}_j) / \sqrt{d} \right),$$

где  $W^k$  — весовая матрица ключа,  $W^q$  - весовая матрица запроса, а  $W^v$  - весовая матрица значения. Такую процедуру внедрения обновлений можно повторить несколько раз, указав количество уровней энкодера.

Уровень прогнозирования. Обновленные представления обрабатываются одним полносвязным слоем, чтобы получить окончательные представления для каждой метки  $\{\mathbf{e}_j^{(final)}\}_{j=1}^K$ . Таким образом, мы получаем  $\{f_j\}_{j=1}^K$ , которые используются в классификаторе с несколькими метками с пороговым значением  $t_j$ , выбранным отдельно для каждой метки с использованием валидационной выборки.

## 4 Планирование эксперимента

### 4.1 Описание данных

Набор данных о продажах [18] — это исторические данные о продажах в разных магазинах. Метки относятся к категориям товаров, а сумма — это количество проданных товаров для определенной категории.

Набор данных о спросе [19] описывает исторический спрос на продукцию нескольких складов. Функция метки означает категорию продукта, а функция количества относится к соответствующему спросу.

Общая статистика по рассмотренным наборам данных приведена в таблице 1.

Мы представляем количество наблюдаемых событий, средний размер набора всех доступных наборов меток  $\text{median}(|Y_{ij}|_{i,j=1,1}^{n,t_i})$ , максимальный размер набора меток, который встречается в набор данных  $\text{max}(|Y_{ij}|_{i,j=1,1}^{n,t_i})$ , количество уникальных меток  $K$  и Diff. Diff измеряет дисбаланс меток: мы вычисляем 5% и 95% квантилей для частот меток и берем разницу между ними. Представление меток не сбалансировано в большинстве наборов данных, но мы используем метрики, которые учитывают этот эффект.

## 5 Метрики

В статье [20] представлен обзор показателей для классификации по нескольким меткам. Мы рассматриваем наиболее часто используемые и всеобъемлющие показатели для оценки качества классификации с использованием нескольких меток.

Каждому образцу  $\mathbf{X}_i$  присваивается набор  $\mathbf{Y}_i$ . Мы можем построить вектор  $\mathbf{y}_i$  из  $\mathbf{Y}_i$  таким образом, что  $y_{ij} = 1$  ( $i \in \{1, \dots, N\}, j \in \{1, \dots, K\}$ ) соответствует случай, когда  $j$ -я метка релевантна для  $i$ -го экземпляра и  $y_{ij} = 0$  в случае нерелевантности. Поскольку модель  $g$  предоставляет окончательный набор меток  $\hat{\mathbf{Y}}_i$  из оценок достоверности  $\mathbf{f}_i$  ( $i \in \{1, \dots, N\}$ ), давайте обозначим вектор прогнозируемых меток как  $\mathbf{g}_i$ . Элемент  $g_{ij}$  равен 1, если сеть прогнозирует  $j$ -ю метку как соответствующую для  $i$ -го экземпляра, и равен 0 в противном случае.

- Micro-F1:

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

- Macro-F1:

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

- Micro-AUC:

$$\text{micro-AUC} = \frac{|W_{\text{micro}}|}{(\sum_{i=1}^N |Y_i^+|)(\sum_{i=1}^N |Y_i^-|)},$$

$$W_{\text{micro}} = \{(a, b, i, j) | (a, b) \in Y_i^+ \times Y_i^-, f_i(x_a) \geq f_j(x_b)\},$$

где  $Y_i^+ = \{j | y_{ij} = 1\}$  и  $Y_i^- = \{j | y_{ij} = 0\}$  обозначают релевантные и нерелевантные метки для  $i$ -го экземпляра соответственно.  $Y_j^+ = \{i | y_{ij} = 1\}$  и  $Y_j^- = \{i | y_{ij} = 0\}$  обозначают наборы положительных и отрицательных экземпляров  $j$ -я метка.

- Macro-AUC:

$$\text{macro-AUC} = \frac{1}{K} \sum_{j=1}^K \frac{W_{\text{macro}}^j}{|Y_j^+| |Y_j^-|}$$

$$W_{\text{macro}}^j = \{(a, b) \in Y_j^+ \times Y_j^- | f_j(x_a) \geq f_j(x_b)\}$$

## 6 Основной эксперимент

Эксперимент проводится на двух выборках Sales и Demand. Данные выборки представлены в таблице 1. Основным результатом будет сравнение подхода с собственным вниманием между представлениями на основе агрегации по меткам классов и подходом, использующим временную агрегацию для представлений. Также будут представлены базовые подходы, работающие с временными рядами это градиентный бустинг и LSTM. Сравнение представлено в Таблице 2.

Dataset	Model	Micro-AUC↑	Macro-AUC↑	Micro-F1↑	Macro-F1↑
Sales	LSTM	<u>0.8670</u>	<u>0.7346</u>	<u>0.5600</u>	<u>0.4389</u>
	TransformerBase	0.8604	0.7206	0.5348	0.3751
	CLASS2C2AE	0.8528	0.6881	0.5116	0.4217
	LANET (ours)	0.9069	0.7627	0.6235	0.4901
Demand	LSTM	0.8829	0.7633	<u>0.6746</u>	0.5929
	TransformerBase	0.8624	0.7240	0.6491	0.5678
	CLASS2C2AE	0.8342	0.7079	0.6738	0.5581
	LANET (ours)	<u>0.8806</u>	<u>0.7373</u>	0.7038	<u>0.5908</u>

Таблица 2: Сравнение нашего метода LANET с базовыми данными по двум различным наборам данных. Выделены лучшие значения, а значения вторые по рангу подчеркнуты.

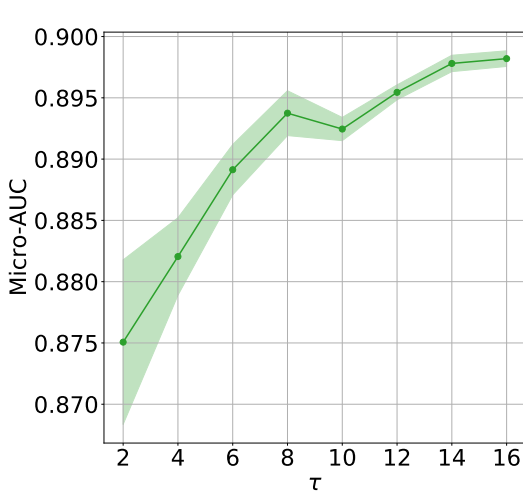
Стоит изучить, какой подход к вычислению собственного внимания более важен: между метками или между временными метками. Таблица 3 отвечает на этот вопрос. Label-attention - это наша базовая

	Micro-AUC↑	Macro-AUC↑	Micro-F1↑	Macro-F1↑
Label-attention	<u>0.881</u> $\pm$ 0.007	<u>0.737</u> $\pm$ 0.017	<u>0.704</u> $\pm$ 0.018	0.591 $\pm$ 0.003
Time-attention	0.837 $\pm$ 0.004	0.682 $\pm$ 0.007	0.674 $\pm$ 0.009	0.588 $\pm$ 0.002
Concat-attention	0.835 $\pm$ 0.001	0.681 $\pm$ 0.010	0.666 $\pm$ 0.000	0.587 $\pm$ 0.001
Gated-attention	0.829 $\pm$ 0.030	0.668 $\pm$ 0.048	0.672 $\pm$ 0.010	0.578 $\pm$ 0.011
Absence-indication	0.882 $\pm$ 0.004	0.742 $\pm$ 0.003	0.704 $\pm$ 0.010	<u>0.589</u> $\pm$ 0.004

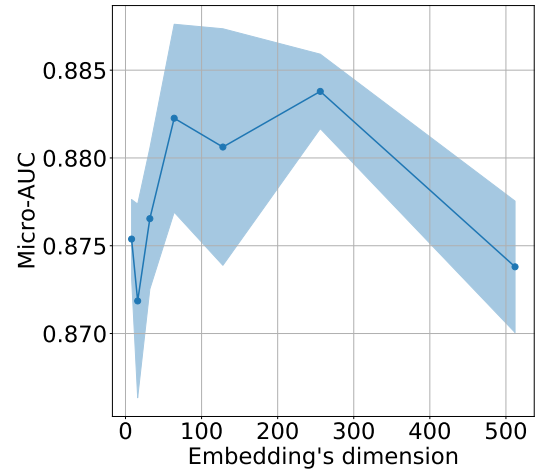
Таблица 3: Сравнение показателей при расчете посещаемости между различными видами в LANET.

реализация. Time-attention - это случай, когда мы учитываем внимание только между просмотрами временных меток. Concat-attention подразумевает получение оценок достоверности путем объединения label-attention и time-attention. Мы узнаем коэффициенты важности двух форм внимания и используем их в качестве весов при суммировании просмотров внимания в случае gated-attention. Эксперимент с индикацией отсутствия заключается в добавлении обучаемого вектора к входным вложениям, если конкретная метка не участвует в рассматриваемой истории.

Важным параметром нашей работы является ограничение исторической информации с помощью параметра  $\tau$ . В нашей работе мы концентрируемся на событиях с временными метками. Модель учитывает только последние временные метки для прогнозирования. Таким образом, возникает естественный вопрос, как качество модели зависит от количества временных меток, используемых для построения прогноза. Длина входной последовательности для модели равна параметру look\_back. Зависимость показателя micro-AUC от look\_back представлена в 2



(a) Зависимость micro-AUC от параметра  $\tau$ .



(b) Зависимость micro-AUC от размера представлений.

Рис. 2: Зависимость показателей от параметров модели для набора данных о спросе.

## Список литературы

- [1] Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor Tsang. The emerging trends of multi-label learning. *IEEE transactions on pattern analysis and machine intelligence*, 2021.
- [2] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 647–657, 2019.
- [3] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. Label-specific document representation for multi-label text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 466–475, 2019.
- [4] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965, 2021.
- [5] Thomas Hartvigsen, Cansu Sen, Xiangnan Kong, and Elke Rundensteiner. Recurrent halting chain for early multi-label classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1382–1392, 2020.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [7] Stephen Grossberg. Recurrent neural networks. *Scholarpedia*, 8(2):1888, 2013.
- [8] Wenyu Zhang, Devesh K Jha, Emil Laftchiev, and Daniel Nikovski. Multi-label prediction in time series data using deep neural networks. *arXiv preprint arXiv:2001.10098*, 2020.
- [9] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.
- [10] Xiao Shou, Tian Gao, Shankar Subramaniam, Debarun Bhattacharjya, and Kristin Bennett. Concurrent multi-label prediction in event streams. In *AAAI Conference on Artificial Intelligence*, 2023.
- [11] Le Yu, Zihang Liu, Leilei Sun, Bowen Du, Chuanren Liu, and Weifeng Lv. Continuous-time user preference modelling for temporal sets prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [12] Fan Zhang, Shuai Wang, Yongjie Qin, and Hong Qu. Conv-based temporal sets prediction for next-basket recommendation. In *2023 International Conference on Frontiers of Robotics and Software Engineering (FRSE)*, pages 419–425. IEEE, 2023.
- [13] Mozdeh Ariannezhad, Ming Li, Sebastian Schelter, and Maarten de Rijke. A personalized neighborhood-based model for within-basket recommendation in grocery shopping. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 87–95, 2023.
- [14] V Ramanjaneyulu Yannam, Jitendra Kumar, Tejaswini Vankayala, and Korra Sathya Babu. Hybrid approach for next basket recommendation system. *International Journal of Information Technology*, 15(3):1733–1740, 2023.
- [15] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.
- [16] Tengshuo Song, Feng Guo, Haoran Jiang, Wenyun Ma, Zhenbao Feng, and Lei Guo. Hgat-br: Hyperedge-based graph attention network for basket recommendation. *Applied Intelligence*, 53(2):1435–1451, 2023.
- [17] Luuk Van Maasakkers, Dennis Fok, and Bas Donkers. Next-basket prediction in a high-dimensional setting using gated recurrent units. *Expert Systems with Applications*, 212:118795, 2023.
- [18] 1C Company. Predict Future Sales. <https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data>, 2018.
- [19] FELIXZHAO. Forecasts for Product Demand. <https://www.kaggle.com/datasets/felixzhao/productdemandforecasting>, 2017.
- [20] Xi-Zhu Wu and Zhi-Hua Zhou. A unified view of multi-label performance measures. In *International Conference on Machine Learning*, pages 3780–3788. PMLR, 2017.

## 7 Выводы

Была рассмотрена задача последовательной классификации с несколькими метками. Для решения этой задачи мы предлагаем архитектуру LANET. В ней исследуется связь между предыдущими векторами с несколькими метками. В частности, LANET использует собственное внимание между метками, чтобы способствовать выявлению исторических взаимозависимостей между ними. LANET демонстрирует наилучшие показатели по двум рассмотренным наборам данных. Однако нынешний подход ограничен случаем, когда для прогнозирования не требуется долговременная память, поскольку мы объединяем только несколько недавних временных меток и общие представления в конкретную последовательность.