Label Attention Network for Temporal Sets Prediction: You Were Looking at a Wrong Self-Attention

Elizaveta Kovtun[a, b,*], **Galina Boeva**[a, c], **Andrey Shulga**[a, c] **and Alexey Zaytsev**[a, d]

[a]Skolkovo Institute of Science and Technology
[b]Sber AI Lab
[c]Moscow Institute of Physics and Technology
[d]Risk Management, Sber

Abstract.

Most user-related data can be represented as a sequence of events associated with a timestamp and a collection of categorical labels. For example, the purchased basket of goods and the time of buying fully characterize the event of the store visit. Anticipation of the label set for the future event called the problem of temporal sets prediction, holds significant value, especially in such high-stakes industries as finance and e-commerce. A fundamental challenge of this task is the joint consideration of the temporal nature of events and label relations within sets. The existing models fail to capture complex time and label dependencies due to ineffective representation of historical information initially. We aim to address this shortcoming by presenting the framework with a specific way to aggregate the observed information into time- and set structure-aware views prior to transferring it into main architecture blocks. Our strong emphasis on input arrangement facilitates the subsequent efficient learning of label interactions. The proposed model is called Label-Attention NETwork, or LANET. We conducted experiments on four different datasets and made a comparison with four established models, including SOTA, in this area. The experimental results suggest that LANET provides significantly better quality than any other model, achieving an improvement up to 65% in terms of weighted F1 metric compared to the closest competitor. Moreover, we contemplate causal relationships between labels in our work, as well as a thorough study of LANET components' influence on performance. We provide an implementation of LANET to encourage its wider usage.

1 Introduction

Numerous domains, such as banking, the grocery industry, etc., treat data as event sequences. For example, in the financial industry, much attention is paid to the history of human banking transactions [7, 8] or the history of purchases in e-commerce [30]. A common problem for event sequences is the prediction of the label for the next event based on the available history [15, 30].

A natural extension of event sequences is temporal set data. For them, we observe a series of timestamped sets, where each set is composed of an arbitrary number of labels, see Figure 1. A primary goal is to predict the next set of labels. The difficulty lies in simultaneously accounting for the temporal sequential behavior of events and labels' interdependencies within sets. Understanding the composition of an expected event allows one to plan more accurately and, as a result, better manage resources.

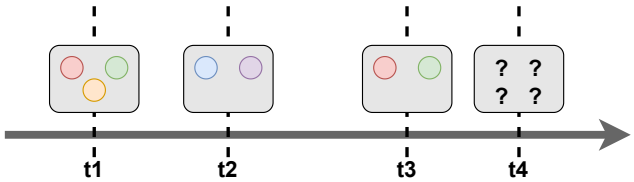

Figure 1: Visual representation of **temporal sets prediction problem**. The sequence of events that are characterized by timestamps t_1, t_2, t_3 and an arbitrary number of labels denoted with colored circles. Our goal is to predict label set for the next event based on the previous sets.

Generally, the multi-label classification is a more natural setting than a binary or multiclass classification since everything that surrounds us in the real world is usually described with multiple labels [21]. There are numerous approaches to deal with the multi-label classification in computer vision [11], natural language processing [40], or classic tabular data domains [32]. Temporal sets prediction can be viewed as multi-label classification problem for consecutive events.

The interaction between an object's states at different timestamps assists in solving tasks with sequential data [13]. Therefore, expressive and powerful models should be able to learn such interactions. Several neural network architectures, such as transformers [35] or recurrent neural networks [9], can do this. For example, a transformer directly defines an attention mechanism that measures how different timestamps in a sequence are connected. However, the applications of modern deep learning methods are limited [50], and they primarily focus on predicting labels for a sequence in general.

We refer to the graph of connections between states of an object at different timestamps as *a timestamp interaction graph*. Another connection worth exploring is the connection between different labels and a need to consider the correlation between them [12]. This capability is absent in the majority of models. We name the graph of connections between different labels *a label interaction graph*.

In our research, we take into account both interaction between labels and timestamped events [17, 44]. For temporal sets prediction, simultaneous consideration of both *timestamp interaction graph* and *label interaction graph* is crucial. Typically, articles explore only one

* Corresponding Author. Email: elizaveta.kovtun@skoltech.ru

Table 1: Mean rank for different metrics averaged over 4 considered datasets. We want to minimize rank, as the best method has a rank of 1. F1 and ROC AUC metric without specification refers to Weighted F1 and Weighted ROC AUC. We denote Hamming Loss as H Loss. The best values are in **bold**, and the second best values are underlined.

Model	Micro F1	Macro F1	F1	ROC-AUC	H Loss
SFCNTSP	4.75	4.75	4.75	4.00	4.0
DNNTSP	3.75	3.75	3.75	3.75	3.0
GPTopFreq	3.00	2.75	3.00	4.25	3.5
TCMBN	2.50	2.75	2.50	2.00	**1.75**
LANET	**1.00**	**1.00**	**1.00**	**1.00**	2.25

side of the dependence that can be explainable by domain bias. In sequential recommendation systems, there is a focus on connections between labels [25] with the incorporation of convolutional neural networks [31] as well as the attention mechanism [49]. Direct models for event sequences [14] prefer the identification of interactions between timestamps [51], considering *a timestamp interaction graph*.

Our LANET aims at conjugate recovery of *label interaction graphs* and *a timestamp interaction graph*, as we believe it is a key moment for modeling temporal sets. The algorithm aggregates past information in specially constructed representations. This aggregation phase is a distinctive feature of LANET that enables it to stand out among others. The built views serve as input to a transformer encoder. The encoder updates embeddings via self-attention, promoting learning of time and label interactions. Finally, we predict a vector of confidence scores for the next-event set based on the model output that encompasses deep knowledge of label relationships. Moreover, we can process long sequences this way, as now the attention evaluation is quadratic in the number of labels, not the sequence length.

Contributions. We propose a transformer-based architecture, called LANET, to effectively deal with temporal sets predictions. Our main contributions are the following:

- We introduce LANET architecture for predicting a label set for the next event, taking the information from previous events. The architecture's peculiarity is based on the specific preparation of the historical information before transferring it into the block with self-attention. The scheme of our approach is presented in Figure 2.
- We conduct a comprehensive comparison of LANET with the well-proven existing models for temporal sets prediction. All experiments indicate that LANET, due to its sophisticated input arrangement, outperforms all considered models by a large margin. See Table 1 for a high-level comparison of different approaches.
- We study the influence of LANET components on its performance. The results suggest that LANET concentrates more on label linkages, while temporal information is in second place by importance.

2 Related Work

Temporal sets prediction resembles a multi-label problem. The multi-label classification problem statement emerges in many diverse domains, e.g., text categorization or image tagging, all of which entail their peculiarities and challenges. The review [48] explores foundations in multi-label learning, discussing the well-established methods as well as the most recent approaches. Emerging trends are covered in a review [21].

We have identified several of the most relevant parts when studying this area. These sections describe significant features and approaches in the most detailed way. Firstly, we examine loss functions tailored for the multi-label setting and some methods for composing label set prediction. Secondly, we overview the usage of RNNs in the multi-label classification task. Thirdly, we review how to capture label dependencies. Then, we discuss an association with a sequential recommendation problem and next basket recommendation.

Loss functions and ways for label set composition in multi-label problem. The paper [23] studies the theoretical background for main approaches to reducing a multi-label classification problem to a series of binary or multi-class problems. In particular, they show that considered reductions are implicitly optimized for either Precision@k or Recall@k. The choice of the correct reduction should be based on the ultimate performance measure of interest. In [19], the authors propose an improved loss function for pairwise ranking in a multi-label image classification task that is easier to optimize. Also, they discuss an approach based on the estimating of the optimal confidence thresholds for the label decision part of the model that determines which labels to include in the final prediction. The task of multi-label text classification is the topic of [10]. The authors construct an end-to-end deep learning framework called ML-Net. ML-Net consists of a label prediction network and a label count prediction network. In order to get the final set of labels, confidence scores generated from the label prediction network are ranked, and then the top K_{top} labels are predicted. A separate label count network predicts K_{top}.

Neural networks for multi-label classification. In [42], the authors use the RNN model to solve a multi-label classification problem. The authors propose to dynamically order the ground truth labels based on the model predictions, which contributes to faster training and alleviates the effect of duplicate generation. In turn, [33] considers the transforming of a multi-label classification problem into a sequence prediction problem with an RNN decoder. They propose a new learning algorithm for RNN-based decoders that does not rely on a predefined label order. Consequently, the model explores diverse label combinations, alleviating the exposure bias. The work [28] examines the same problem statement of multi-label classification in an event stream as we do. The authors' model targets capturing temporal and probabilistic dependencies between concurrent event types by encoding historical information with a transformer and then leveraging a conditional mixture of Bernoulli experts. This article [45] discusses the formulation of the task of predicting time sets for users; it offers a continuous learning system that allows you to explicitly capture changing user preferences by maintaining a memory bank that could store the states of all users and items. In this paradigm, the authors construct a non-decreasing universal sequence containing all user-defined interactions, then chronologically learn from each interaction. To research the cross-relation between products in the basket, a ConvTSP [47] was proposed that combines dynamic user interests and statistical interests into a single vector representation for a user.

Approaches to leveraging label dependencies. The authors of [16] construct a model called C-Tran for a multi-label image classification task that leverages Transformer architecture that encourages capturing the dependencies among image features and target labels. The key idea is to train the model with label masking. The authors in [43] propose DNN architecture for solving the multi-label classification task, which incorporates the construction of label embeddings with feature and label interdependency awareness. A label-correlation sensitive loss improves the efficiency of the constructed model. Another popular way to consider label relationships is to use Graph Neural Networks as a part of the pipeline. Namely, [24] captures the correlation between the labels in the task of Multi-Label

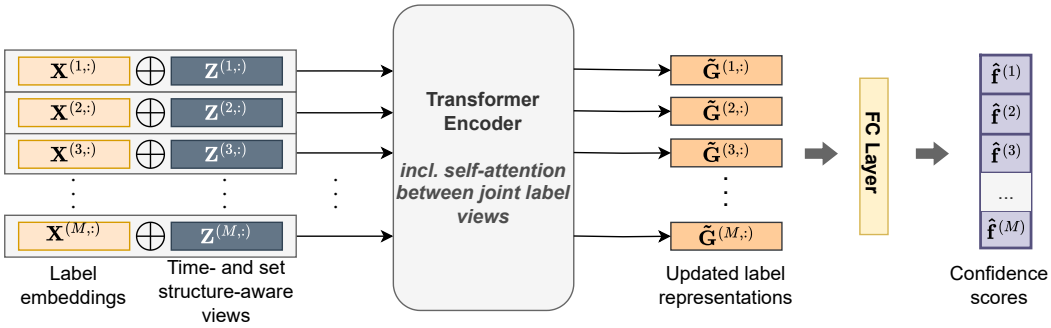

Figure 2: LANET architecture for temporal sets prediction. The key part is to aggregate historical information into representative views that will be transferred into the Transformer encoder block. The output of the model is a vector of confidence scores, whose components are associated with the prospect of a corresponding label to be a member of the next-event set.

Text Classification by adopting a Graph Attention Network (GAT). They predict the final set of labels combining feature vectors from BiLSTM and attended label features from GAT. Event sequence processing also tries to derive dependencies between different event types and consider specific attention mechanisms [22]. The closest to our LANET approach [20] explores the relationship between different series for multivariate time series classification. The authors propose using attention in step-wise and channel-wise fashion to produce embeddings, which are then forwarded by a classification head.

Sequential recommendation systems. One more close neighbor of our problem statement is the problem of sequential recommendation system construction [25, 37]. In this case, we have many possible labels, and we should sort them by probability of occurrence next time. Typically, the estimation of embeddings for all possible labels/items is a part of a pipeline. Existing approaches use neural networks for sequential data such as LSTM [38] as well as attention mechanism [36]. We want to highlight the statement related to the usage of only recent past data for prediction [17]. However, millions of possible labels typically lead to more classic techniques in this area with specific loss functions and methods.

Next basket recommendation. The next relevant problem is the next basket recommendation. This formulation is similar to ours, so we also considered many approaches and ideas when analyzing our research area. The authors in [6] proposed a personalized model that captures short-term dependencies within a temporary set of products, as well as a long-term one based on historical user information. Also, in [41], to connect local and global user information, a hybrid method based on an autoencoder for context extraction and RNN for understanding the dynamics of changing interests is proposed. To overcome similar problems, a graph-based hyperedge-based attention network [29] is being created for the following recommendation. In this formulation of the problem, there is difficulty working with a dictionary of product categories since they number thousands of values; [34] uses GRU to predict the next basket, which is easily scaled to a large assortment.

3 Methodology

This section presents the formalization of the temporal sets prediction problem and the description of our LANET method that addresses it effectively. The overall architecture of LANET is presented in Figure 2. We expand in LANET parts consolidation of historical information into joint label representations, application of Transformer encoder, and obtaining a vector of confidence scores.

3.1 Temporal Sets Prediction

In event sequence theory, each event is characterized by one categorical label and a timestamp. In practice, there are lots of available event sequences related to different users with their underlying development patterns. When dealing with such data structure, the widespread goal is to capture user- and general-level hidden sequence regularities to predict future behavior. Mostly, an event is attributed not with a single label but with some set of labels. It is a more general and realistic problem statement to consider the possibility of a time moment being concurrently associated with various marks. For instance, engaging a number of services in the app, purchasing several items in the online store, or conducting various kinds of transactions over some period of time. Therefore, the transition from temporal events to temporal sets can be viewed as an act of generalization. In what follows, we treat **Temporal Sets** as a sequence of event-related timestamped sets composed of an arbitrary number of labels. In turn, **Temporal Sets Prediction** is a problem of predicting a label set tied to the next event on the basis of an observed sequence of event-associated sets.

The problem of temporal sets prediction can be formalized as follows. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ be the collection of N users. Each user $i, 1 \leq i \leq N$, is bound with a sequence of temporal sets $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$, where T is a number of the observed timestamps. A set $s_i^j, 1 \leq i \leq N, 1 \leq j \leq T$, is a collection of an arbitrary number of labels sampled from a vocabulary $\mathcal{Y} = \{y_1, y_2, \ldots, y_M\}$ of size M. Given a sequence of historical sets $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ for user $u_i \in \mathcal{U}$, where each set $s_i^j \subset \mathcal{Y}$, the goal of temporal sets prediction problem is to predict the subsequent label set $\hat{s}_i^{T+1}$, that is,

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

where $\mathbf{W}$ relates to trainable parameters of function g. Function g should be able to grasp the consecutive development of sets in a sequence $\mathcal{S}_i$ as well as label interaction within each set s_i^j.

3.2 Our LANET approach

The principal aspects of the temporal sets prediction problem are the time-evolving nature of set series and the complex inner organization of individual sets. Notably, these peculiarities are interconnected and complementary, requiring a joint record. Mindful of the importance of their concurrent treatment, we propose a model LANET that is targeted at such a challenge. In particular, we propose to calculate self-attention between specifically designed representations of

historical information. Such representations encompass the knowledge of the time of the events happening and the label composition of each event-related set. The usage of the self-attention mechanism over constructed representations enables the identification of time- and label-aware relationships. Finally, we apply affine transformations to the updated representations at the output of self-attention to get a vector of confidence scores for the next event labels.

Representation of historical information in LANET. First of all, we want to effectively aggregate past information on event times and set structures for the sequence $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$. Let $\mathbf{X} \in \mathbb{R}^{M \times D}$ denote the embedding matrix of all labels from the vocabulary $\mathcal{Y} = \{y_1, y_2, \ldots, y_M\}$, where D is a dimension of embedding vectors. The parameters of the matrix $\mathbf{X}$ are initialized from the standard normal distribution and later updated in the training process. An important step is the construction of time representations. Each set s_i^j is connected with time t_j. The countdown of time starts from one common point for all users. For each timestamp $j, 1 \leq j \leq T$, we establish temporal embedding $\mathbf{t}_j \in \mathbb{R}^D$, as it is done in [28]:

$$\mathbf{t}_j^{(d)} = \begin{cases} \cos\left(t_j / 10000^{\frac{d-1}{D}}\right), & \text{if } d \text{ is odd,} \\ \sin\left(t_j / 10000^{\frac{d}{D}}\right), & \text{if } d \text{ is even,} \end{cases}$$

where $d, 1 \leq d \leq D$, is a component of a vector of dimension D. After defining representation for each time moment $t_j, 1 \leq j \leq T$, we aggregate all time-related knowledge from sequence $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ into matrix $\mathbf{Z} \in \mathbb{R}^{M \times D}$. The m-th row, $1 \leq m \leq M$, of matrix $\mathbf{Z}$, denoted as $\mathbf{Z}^{(m,:)}$, is equal to the sum of embeddings of timestamps, in which label $y_m \in \mathcal{Y}$ appears as a member of set:

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

If label y_m is not encountered in any set of the sequence $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$, then the m-th row of matrix $\mathbf{Z}$ will consist of all zeros. Hence, in the case of meeting label y_m in several sets of the sequence $\mathcal{S}_i$, the corresponding m-th row of matrix $\mathbf{Z}$ will be the sum of all relevant time embeddings for this particular label.

The united representation of sequence $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ is a concatenation of defined matrices, embodying time and set structure information:

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

The rows of resulting matrix $\mathbf{G} \in \mathbb{R}^{M \times 2D}$ are regarded as joint representations of corresponding labels. Namely, m-th row of matrix $\mathbf{G}$ is a joint view of label y_m. The designed representation of each label includes its view, expressed in $\mathbf{X}$, and part responsible for time-aware interaction with other labels, found in $\mathbf{Z}$.

Learning relations via self-attention in LANET encoder. We define the joint label representations as rows of matrix $\mathbf{G}$, which involve self-oriented label information as well as time-aware knowledge of label interrelationships. For the encouragement of further relation capturing, we apply the self-attention mechanism over the matrix $\mathbf{G}$ to get its updated version $\tilde{\mathbf{G}}$:

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

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are query, key, and value matrices, which are linear transformations of matrix $\mathbf{G}$. The main block of LANET architecture consists of several transformer encoder layers with multi-head self-attention. Leveraging self-attention, we fuse historical records expressed through joint-label views and emphasize essential interactions. The updated label representations are infused with retrospective time- and set structure-aware information.

LANET prediction layer. Finally, the updated representations $\tilde{\mathbf{G}} \in \mathbb{R}^{M \times 2D}$ take part in obtaining the confidence scores for all labels to be included in the next-event set:

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

where $\hat{\mathbf{f}} \in \mathbb{R}^M$ is a confidence score vector of size of label vocabulary, $\mathbf{W}^{\text{out}} \in \mathbb{R}^{2D \times 1}$ and $b^{\text{out}} \in \mathbb{R}$ are trainable parameters of the prediction layer. We use the sigmoid activation function to make confidence scores lie in the $[0, 1]$ range. Therefore, the m-th component, $1 \leq m \leq M$, of confidence vector $\hat{\mathbf{f}} \in \mathbb{R}^M$ is associated with a prospect of label y_m to become a part of the predicted set $\hat{s}_i^{T+1}$.

LANET learning process. The output of the LANET prediction layer is a confidence score vector $\hat{\mathbf{f}} \in \mathbb{R}^M$. Vector $\hat{\mathbf{f}}$ provides the basis for predicting the composition on the next-event set $\hat{s}_i^{T+1}$. For model training and validation, we use a real next set s_i^{T+1} as a ground truth. LANET is trained in end-to-end fashion, taking the historical sequence $\mathcal{S}_i = \{s_i^1, s_i^2, \ldots, s_i^T\}$ as an input and producing a vector of confidence score $\hat{\mathbf{f}}$ as an output. We adopt the following loss function:

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

where $\mathbf{I}_m = \mathbf{I}\{y_m \in s_i^{T+1}\}$ is an indicator function of label y_m to be a member of a set s_i^{T+1}, while $\mathbf{I}_m^{'}$ is an indicator function with the opposite condition $\mathbf{I}_m^{'} = \mathbf{I}\{y_m \notin s_i^{T+1}\}$. We denote the m-th component of the predicted confidence score vector $\hat{\mathbf{f}}$ as $\hat{\mathbf{f}}^{(m)}$. The formula of loss function $\mathcal{L}_i$ is given for the case when we consider only one user u_i. In view of all available users $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, we minimize the sum of all user-related loss components $\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_i$ in the training process.

In the training dataset for LANET, the bundle of the set s_i^{T+1} as a ground truth and a sequence $\{s_i^1, s_i^2, \ldots, s_i^T\}$ as input is not the only one training example that is drawn from the user sequence $\mathcal{S}_i$. To increase the amount of training data, we also leverage all intermediate sets in a sequence as a ground truth and preceding sets as the model's input. Thus, $s_i^j, 2 \leq j \leq T+1$, are taken as target sets and the subsequences $\{s_i^1, \ldots, s_i^{(j-1)}\}$ as corresponding inputs.

4 Experiments

In this section, we present the performance comparison of our LANET approach with the existing models for temporal sets prediction problem. Besides, we perform a thorough ablation study that reveals insights into LANET working details. The code for LANET is available at GitHub repository[1].

4.1 Datasets

After analysis of the works devoted to models for temporal sets prediction, we identify four frequently used datasets:

- **Dunnhumby-Carbo (DC)** [4]: This dataset includes transactional data of households in a retail store over two years. Here, sets are products assigned to one transaction.

[1] https://github.com/adenshulga/LANET

Table 2: Comparison of our LANET approach with the existing models for temporal sets prediction on four datasets. Best values are highlighted, and second-best values are underlined.

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**

- **Mimic III** [2]: It consists of the medical records for patients from intensive care. The patient-related event constitutes a hospital admission time and a set of disease classification codes.
- **Instacar**t [3]: The Instacart dataset comprises records of users' product orders. Each event is described by a time of purchase and a set of product labels.
- **Synthea** [5]: This is synthetically generated EHR data with simulated medical events, similar to the MIMIC III dataset.

Statistics of these datasets are given in Table 3. We provide the overall number of sets in each dataset (#Sets), the median set size (MdnSS), the maximum set size (MaxSS), the size of label vocabulary (Vocab), the mean length of historical sequences (MnLen), and the number of available sequences (#Seqs).

Table 3: 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

4.2 Compared Methods

The following methods are compared with our LANET approach:

- **GPTopFreq** is a frequency-based baseline, inspired by [18]. This method evaluates the frequencies of each label occurrence in the whole dataset and in the user-related history. Then, for each label, GPTopFreq takes the maximum of "general" and "personal" frequencies and uses it as the predicted probability.
- **DNNTSP**[2] model is described in [50]. DNNTSP constructs a co-occurrence frequency graph, performs weighted graph convolutions on it to learn element relationships, utilizes an attention-based module to learn the temporal dependency of elements in sets, and uses a gating mechanism to fuse static and dynamic information about elements.
- **SFCNTSP**[3] is a model for temporal sets prediction presented in [46]. It comprises four consequent modules, namely Simplified Fully-Connected Networks, that learn inter and intra-set dependencies, intra-embedding channel correlations, and user representations.
- **TCMBN**[4] model idea is given in [28]. TCMBN leverages Transformer-based architecture to capture probabilistic dependency between elements in sets via neural density estimation of parameters of Bernoulli mixture and temporal dependency between sets via attention.

We take these models because they are pretty recent in temporal sets prediction and demonstrate high performance. Their hyperparameters for different datasets are set to the values specified by the authors.

4.3 Implementation details

Our LANET model consists of several transformer encoder layers with multi-head self-attention. The number of layers is equal to 2 in all cases, while the number of self-attention heads ranges from 4 to 6, depending on the particular dataset. As a basis, we take the Transformer layer implementation from PyTorch [1]. We apply dropout with the probability of 0.2 directly to the output of the transformer encoder block. LANET quality dependence on the model hyperparameters will be presented in Section 4.6. For the training procedure, we use the Adam optimizer with an initial learning rate of 0.001. For the scheduler, we adopt "reduce on Plateau" strategy with patience of 10 epochs and factor of 0.9.

4.4 Validation metrics

The original datasets are divided into train, validation, and test sets. Splits are performed on users. Thus, time periods in the train, valid, and test parts overlap. We take 60% of the data samples for model training, 20% for validation, and 20% for testing. All experiments are launched with five different random seeds; the mean and standard deviation of the results are calculated.

Evaluation of temporal sets prediction is similar to validation of multi-label classification, so we use well-established and comprehensive metrics from multi-label domain [39] and metrics from the

[2] https://github.com/yule-BUAA/DNNTSP
[3] https://github.com/yule-BUAA/SFCNTSP
[4] https://github.com/xshou1990/TCMBN

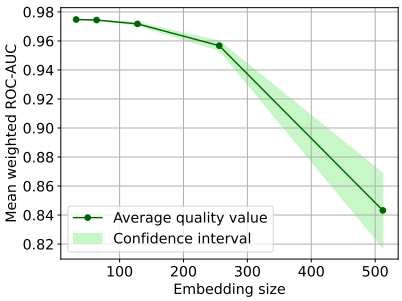

Figure 3: The dependence of LANET quality on the embedding size.

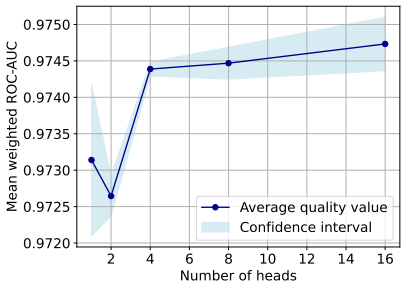

Figure 4: The dependence of LANET quality on the number of heads.

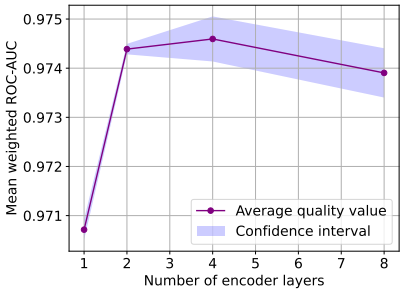

Figure 5: The dependence of LANET quality on the number of encoder layers.

relevant works [28] of the considered area of temporal sets. Thus, we employ Hamming Loss, Weighted ROC-AUC, Weighted F1, Micro F1, and Macro F1 metrics for ultimate quality assessment. Meantime, the calculation of micro-F1, macro-F1, and Weighted F1 implies operation with the predicted label sets, not with the label confidence scores. In this regard, the transition from output scores to the predicted label sets is done by comparison of the label-related confidence scores with certain thresholds. These thresholds are calculated on the validation set by optimization of F1-score for each label separately.

4.5 Main results

Metrics for comparison of our LANET approach with the established models for temporal sets prediction problem are presented in Table 2. LANET demonstrates top-1 performance on all datasets, substantially surpassing its competitors. A huge performance gap is observed on the DC, which can be connected with a vast number of available sequences for training in this dataset or with the specific set structures. The closest competitor for LANET is the TCMBN model, which is also based on transformer architecture. The results indicate that the crucial moment is the treatment of the historical information at the model entrance rather than its processing afterwards. LANET successfully copes with this challenge and shows an absolutely different level of performance. Interestingly, the statistical baseline GP-TopFreq demonstrates a higher quality than the deep neural network models in some cases. Such phenomenon is also mentioned in [18].

4.6 Ablation study

We investigate the dependence of LANET performance on its major hyperparameters. Unless otherwise specified, MIMIC III dataset is under consideration.

Contribution of time information. Each set in a sequence is attributed with a timestamp, which takes part in obtaining time representations. We decide to contemplate the contribution of the time component to model performance. So, we omit temporal information from LANET by substituting time representations with the constant vector. Such a vector indicates the particular label's presence in the user history, neglecting all time dependencies. The metric drops as a result of this modification are given in Table 4. However, even after the exclusion of the time-aware views from LANET, it still demonstrates elevated performance due to the efficient processing of similar frequency-based history representation.

Dependence of LANET performance on embedding size. An essential part of our model is the utilization of learnable embeddings

Table 4: The contribution of temporal information into LANET performance. Metric drops in case of time omission are provided for Weighted F1 and Weighted ROC-AUC.

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

for managing temporal sets. For this reason, it is necessary to examine the influence of embedding dimensionality on LANET metrics because this parameter is directly related to a model capacity. The dimension of joint representations before transferring into the transformer encoder bock equals $2D$. The effect of changing the values of D is presented in Figure 3. From it, we can conclude that LANET struggles to learn representations of high dimensions effectively.

Dependence of LANET performance on number of heads in attention layers. The usage of several heads in the attention layers allows the model to account for multiple distinct dependencies, dedicating an individual head to grasping the specific pattern. Figure 4 confirms that the more significant number of adopted heads leads to quality enhancements. However, resource consumption grows alongside the increase in the head quantity.

Dependence of LANET performance on a number of encoder layers. The hyperparameter of the number of encoder layers is responsible for the capability to recognize complex relationships within data. Figure 5 demonstrates that there exists an optimal number of layers for solving the considered problem. The further increase in the number of layers brings in the failure to train the effective model.

Graph interpretation of attention weights. An essential part of the resulting architecture is the encoder layer, which includes the attention layer. Attention, in turn, indicates the degree of relevance of the relationship between the labels, which is significant for further model prediction. We select the most relevant labels for a selection in Instacart to identify the causal explanations of label predictions. The Figure 6 on the left shows the heatmap for their relationships. We notice that the attention matrix clearly dominates of the labels encountered in the sequence over those that are not in it, which is clearly expressed through the weights. Looking deeper, we see that small-scale variations in attention describe the connection between particular event types.

Furthermore, we consider the most relevant labels for a sampling.

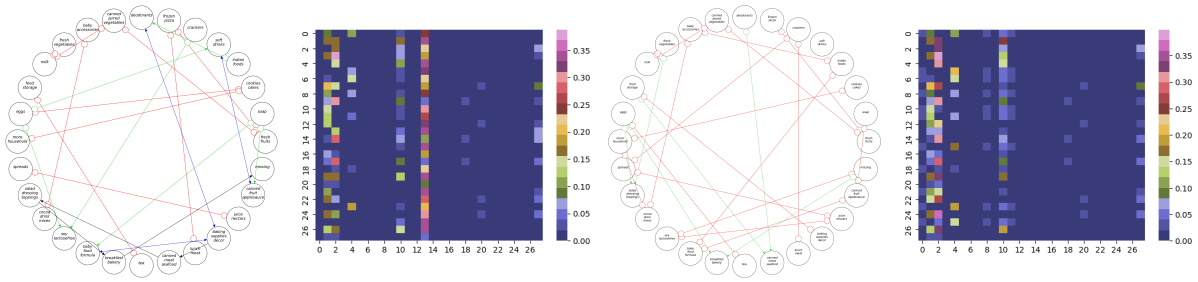

Figure 6: Interpreting the relationship of labels using the attention layer. On the left is a picture showing the relationship between a subset of labels and their verbal interpretation. Next to the graph is a heatmap, which illustrates the relationship of all possible labels of the Instacart dataset. On the right are the modified graphs, which are obtained as a result of removing the label with the highest attention weight in all possible values and the corresponding distribution of weights in the heatmap. The data is obtained from the dataset Instacart.

The figure on the left shows the heatmap for their relationships. To generate causal explanations, we needed a graph visualization of the attention scales for individual labels. This is an idea behind framework CLEANN [27], which proposes a method to extract causal relationships as a partial ancestral graph (PAG) [26]. So, to form the graph, we looked at one of the users and the corresponding historical information about the labels. Using the pretrained LANET model, we obtained the attention weights fed into the CLEANN algorithm.

The left visualization of the graph in Figure 6 has several types of connections:

- The red lines indicate the proximity of the labels inside the graph;
- The blue connections are more complex, this is a bidirectional interaction between the labels in the graph;
- Black means that the label is the parent for the subsequent;
- The greens, on the contrary, are children.

In the first case, complex and intricate relationships between labels have developed. For example, if “canned meat seafood" is the parent, you will generate “salad dressing toppings." Some connections may seem counterintuitive to us, but this story is individual for each user when buying goods in the store.

Moreover, in order to find out and identify the connections, we decided to remove the label with the highest total weight in the attention matrix and look at the redistribution of weights in this case (Figure 6 on the right). The model turned the attention to a variety of other labels. The PAG demonstrates a changed picture, where all the blue and black edges of the graph have disappeared, which corresponds to a more complex and oriented connection than a simple “neighborly" one. The correlation between labels has become lower. Moreover, “canned meat seafood" changed its behavior. It has become a subsidiary and no longer has connections with anyone, which affects the predictive ability of this label for the next time step. This exploration indicates that the best predictive capabilities of LANET mainly depend on the model’s ability to detect relationships between labels rather than on building a work with time and the order in which baskets are placed.

5 Conclusion

In this work, we consider the problem of temporal sets prediction: given the history of timestamped sets comprised of an arbitrary number of categorical labels, the goal is to predict the collection of labels for the next event. To solve this problem, we propose the LANET model. LANET is remarkable for its early effective aggregation of historical information into vector representations, not encountered in other existing models. The specific view on the available information enables further effective capturing of time and label interdependencies. Our method demonstrates the best performance on four reviewed datasets, surpassing SOTA approach and providing improvement of 65% in terms of Weighted F1 on one of the datasets. As for the limitations, LANET shows consistently strong results, specifically on datasets with label vocabulary sizes of $100-200$. The adaptation to the recommendation setting, in which item vocabulary size can reach thousands, or to the setup with the much smaller vocabulary of $5-20$ are open questions. Besides, the issue that is worthy of consideration is the study of effects from the reduction of event sequences to temporal sets. Such contraction can be done by choosing an appropriate time period for group formation but may introduce unexpected findings in event sequence tasks. In addition to, the proposed approach naturally fits into the paradigm of self-supervised learning and can serve as a source of valuable representations for the downstream tasks.

6 Acknowledgments

The research was supported by the Russian Science Foundation grant 20-7110135. The authors would like to thank Evgenia Romanenkova for her careful paper review and valuable advice.

References

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

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

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

[4] Dunnhumby-Carbo. https://www.dunnhumby.com/source-files/, 2020.

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

[6] M. Ariannezhad, M. Li, S. Schelter, and M. 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.

[7] D. Babaev, N. Ovsov, I. Kireev, M. Ivanova, G. Gusev, I. Nazarov, and A. Tuzhilin. Coles: Contrastive learning for event sequences with self-supervision. In *Proceedings of the 2022 International Conference on Management of Data*, pages 1190–1199, 2022.

[8] A. Bazarova, M. Kovaleva, I. Kuleshov, E. Romanenkova, A. Stepikin, A. Yugay, D. Mollaev, I. Kireev, A. Savchenko, and A. Zaytsev. Universal representations for financial transactional data: embracing local, global, and external contexts. *arXiv preprint arXiv:2404.02047*, 2024.

[9] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[10] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, and Z. Lu. Ml-net: multi-label classification of biomedical texts with deep neural networks. *Journal of the American Medical Informatics Association*, 26(11):1279–1285, 2019.

[11] T. Durand, N. Mehrasa, and G. 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.

[12] J.-Y. Hang and M.-L. Zhang. Collaborative learning of label semantics and deep label-specific features for multi-label classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[13] T. Hartvigsen, C. Sen, X. Kong, and E. 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.

[14] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[15] W.-C. Kang and J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[16] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16478–16488, June 2021.

[17] J. Li, Y. Wang, and J. McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330, 2020.

[18] M. Li, S. Jullien, M. Ariannezhad, and M. de Rijke. A next basket recommendation reality check. *ACM Transactions on Information Systems*, 41(4):1–29, 2023.

[19] Y. Li, Y. Song, and J. Luo. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song. Gated transformer networks for multivariate time series classification. *arXiv preprint arXiv:2103.14438*, 2021.

[21] W. Liu, H. Wang, X. Shen, and I. Tsang. The emerging trends of multi-label learning. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[22] H. Mei, C. Yang, and J. Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2021.

[23] A. K. Menon, A. S. Rawat, S. Reddi, and S. Kumar. Multilabel reductions: what is my loss optimising? *Advances in Neural Information Processing Systems*, 32, 2019.

[24] A. Pal, M. Selvakumar, and M. Sankarasubbu. Multi-label text classification using attention-based graph neural network. *arXiv preprint arXiv:2003.11644*, 2020.

[25] M. Quadrana, P. Cremonesi, and D. Jannach. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.

[26] T. Richardson and P. Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

[27] R. Y. Rohekar, Y. Gurwicz, and S. Nisimov. Causal interpretation of self-attention in pre-trained transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] X. Shou, T. Gao, S. Subramaniam, D. Bhattacharjya, and K. Bennett. Concurrent multi-label prediction in event streams. In *AAAI Conference on Artificial Intelligence*, 2023.

[29] T. Song, F. Guo, H. Jiang, W. Ma, Z. Feng, and L. Guo. Hgat-br: Hyperedge-based graph attention network for basket recommendation. *Applied Intelligence*, 53(2):1435–1451, 2023.

[30] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[31] J. Tang and K. Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[32] A. N. Tarekegn, M. Giacobini, and K. Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118: 107965, 2021.

[33] C.-P. Tsai and H.-Y. Lee. Order-free learning alleviating exposure bias in multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6038–6045, 2020.

[34] L. Van Maasakkers, D. Fok, and B. Donkers. Next-basket prediction in a high-dimensional setting using gated recurrent units. *Expert Systems with Applications*, 212:118795, 2023.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] S. Wang, L. Hu, and L. Cao. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 17*, pages 285–302. Springer, 2017.

[37] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun. Sequential recommender systems: challenges, progress and prospects. In *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 6332–6338. International Joint Conferences on Artificial Intelligence, 2019.

[38] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503, 2017.

[39] X.-Z. Wu and Z.-H. Zhou. A unified view of multi-label performance measures. In *International Conference on Machine Learning*, pages 3780–3788. PMLR, 2017.

[40] L. Xiao, X. Huang, B. Chen, and L. 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.

[41] V. R. Yannam, J. Kumar, T. Vankayala, and K. S. Babu. Hybrid approach for next basket recommendation system. *International Journal of Information Technology*, 15(3):1733–1740, 2023.

[42] V. O. Yazici, A. Gonzalez-Garcia, A. Ramisa, B. Twardowski, and J. v. d. Weijer. Orderless recurrent models for multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13440–13449, 2020.

[43] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang. Learning deep latent space for multi-label classification. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[44] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu. Sequential recommender system based on hierarchical attention network. In *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[45] L. Yu, Z. Liu, L. Sun, B. Du, C. Liu, and W. Lv. Continuous-time user preference modelling for temporal sets prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[46] L. Yu, Z. Liu, T. Zhu, L. Sun, B. Du, and W. Lv. Predicting temporal sets with simplified fully connected networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4835–4844, 2023.

[47] F. Zhang, S. Wang, Y. Qin, and H. 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.

[48] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8): 1819–1837, 2013.

[49] S. Zhang, Y. Tay, L. Yao, A. Sun, and J. An. Next item recommendation with self-attentive metric learning. In *Thirty-Third AAAI Conference on Artificial Intelligence*, volume 9, 2019.

[50] W. Zhang, D. K. Jha, E. Laftchiev, and D. Nikovski. Multi-label prediction in time series data using deep neural networks. *arXiv preprint arXiv:2001.10098*, 2020.

[51] V. Zhuzhel, V. Grabar, G. Boeva, A. Zabolotnyi, A. Stepikin, V. Zholobov, M. Ivanova, M. Orlov, I. Kireev, E. Burnaev, R. Rivera-Castro, and A. Zaytsev. Continuous-time convolutions model of event sequences, 2023.