MASTER'S THESIS STATUS REPORT

Addition of External Information for Enhancement of Local Embeddings for Event Sequences Data Models

Master's Educational Program: Data Science

Student: Maria Kovaleva
Advisor: Alexey Zaytsev

Moscow 2023

1 Abstract

Neural network models for representations of sequences of events are especially interesting in the area of bank users' transactions. The large amount of transactional data and its relevance to various problems of describing users' behavior makes such representations an important tool in bank practice.

In this work we take into account external information for a more accurate description of users' current state. An accounting of the external context was performed by using aggregation of the other clients' representations at the current time moment. Another important part of the project was the validation of the developed methods and models. The proposed models were considered in the context of solving applied downstream tasks in the banking sector. Considered tasks included credit scoring and fraud detection.

The obtained results show that the models developed in this project work better than the similar ones but without accounting of external information in terms both global and local properties of representations.

The resulting model for constructing representations will allow to solve a wide range of problems and can be implemented in the bank. Its use will improve the quality of the decision making process and will allow the use of transactional data representations in various applications.

Contents

2 Introduction

Any bank keeps a lot of information about the clients' transactions. It is natural to assume that this big sequential data contains useful for business but hidden information about users. The availability of a large set of such data makes it possible to train various machine learning models [1]. Modern methods specially focus on representation learning models including representations of transactional sequences [2, 3, 4]. The learned embeddings are used later as input in various applied problems [2, 5].

Representations of event sequences for bank clients can have two important properties:

1. Global properties — characterize the client as a whole, at the level of the entire set of users.

2. Local properties — are responsible for describing the state of a specific client at a specific point in time.

Existing approaches for constructing event sequence representations consider only one group of properties, either local, or global information. Thus, methods based on the contrastive approach for constructing a representation consider the whole sequence of events [2, 3]. In this approach the importance of local changes in sequences is blurred and additional techniques are required to overcome this problem. For example, in the work [6] hierarchical loss function is used.

Despite the fact that existing models have some global properties, they do not take into account the external context when forming a representation of transactional data. For example, representations of other bank clients or macroeconomic parameters [7]. Accounting of this information is a new task that will potentially improve the quality of the models [8].

Thus, the purpose of this work is to improve the local and global properties of transaction data representations for a bank user using accounting of external information. The purpose defines the main problems of the project:

1. Development and research of machine learning models to obtain representations of transactional data;

2. Study of the possibility of improvement for the obtained representations by using of external context;

3. Testing the received embeddings to solve several applied problems.

To solve these problems, we develop existing methods for transactional data, in particular CoLES, add accounting of external contex and validate all received representations on a number of applied problems.

Proposed approaches will improve the process of creating machine learning models based on transactional data in the bank, improve the quality of the models and, consequently, the quality of decisions that are made based on forecasts obtained using such models.

3 Literature review

This section is dedicated to reviewing the existing methods of learning event sequence representations which possess the desired properties: efficiency when solving both global and local tasks, and external information awareness.

3.1 Event sequence representation learning

Neural networks have been shown to perform well when faced with the task of event sequence representation learning, both in local and global problems [2]. Local representations are used for tasks, constrained to one event sequence, such as next event prediction [9] or change point detection [10, 8]. On the other hand, global representations may be applied to compare and classify whole sequences [1, 11].

Neural network solutions are known to be very sensitive to the domain they are applied in, so it is crucial to keep in mind the distinctive features of our domain. Firstly, the data we work with is both temporally and spatially correlated, so preprocessing should be done with care [8]. Secondly, transactional data sequences have some key differences with simple time series:

1. time series usually have a uniform distance between events, which is not true for event series;

2. in contrast to the sequences in a time series dataset, the sequences in an event type dataset don't all necessarily have the same length;

3. transactional data has both categorical features (MCC codes), and continuous features (amount).

3.2 Self Supervised Learning Paradigm

Self-supervised learning (SSL) is one of the most more promising methods for obtaining a representation of data from various types. It is based on the fact that the marking of similar and dissimilar objects can be obtained directly from data, including sequential ones, without using external markup or involving an experts. There are two main groups of methods for this approach: contrastive and generative [12].

3.2.1 Contrastive models

Contrastive methods all seek to achieve the following objective: a perfect model would yield close representations for "similar" objects, and distant representations for "dissimilar" objects. Similarity measure derived naturally from the domain of the task. For example, in our case, a pair of slices of the same sequence may be marked similar, and any other pair – dissimilar [2].

Contrastive methods first appeared in the field of computer vision: siamese models with triplet loss [13], SimCLR [14], and later DINO [15], BarlowTwins [16] are all methods, capable of learning high quality image representations.

These methods are also effective when applied to multidimensional event sequences and time series. In [3], the authors have combined neural networks with the Nystrom kernel method, and achieved competitive quality. Note, that [3] views different timestamps as separate objects in terms of contrastive learning, which may improve the representations' generalization ability. Luckily, that is not the case, and such approaches outperform the classical methods of risk evaluation.

The paper [2] proposes the CoLES method, which uses sequence slices as the contrasted objects. This paper also demonstrated, how such representations can be successfully applied to the tasks of gender classification, age regression, fraud detection, and others.

On the other hand, CoLES and some of the other mentioned before models have a bunch of drawbacks. Firstly, CoLES, trained on whole sequences, performs poorly on smaller sequence slices. Secondly, it cannot be used to track the changes in the behavour of a single user: all slices from a common sequence will have close representations. Besides that, it could be argued that similar subsequences need to have close representations, even if they come from different users, which is definetly not the case for CoLES. All in all, CoLES representations show good performance when faced with global, inter-user tasks, but local tasks require a different training procedure.

3.3 Accounting for external information

All the mentioned above methods consider only one sequence at a time. It is sometimes beneficial to consider the global context, formed from the actions of every client [2, 8].

Global context may consist of the behaviour of other specific clients, or it may reflect the current macroeconomic state. It has been shown, that

the two strongly correlate [17, 7]. This implies that the global context may contain information, useful for model training.

Since choosing macroeconomic indicators is difficult without the proper education, and thus requires bringing in an expert in the field, it makes more sense to try different ways to aggregate the actions of all bank clients (mean, max, etc.). This approach allows to extract more data from the dataset, without additional annotation.

3.4 Sequence embedding evaluation

The process of evaluation and comparison of the proposed approaches is a very important part of research, as it allows the researcher to test the applicability of the considered methods. For our task, we require a very specific set of benchmarks, which would test the local, global and dynamic properties of sequence representations.

To evaluate the global properties, we shall use the approach from [2]. The subsequences are sampled from the parent sequence in such a way, that each interval corresponds to a single label. This makes it simple to pose a classification problem on the resulting subsequences. The ability to solve such a problem is postulated as indicative of the global properties of used representations.

For testing the local properties, we shall turn to the classic task of next token prediction, which comes from the field of temporal time processes [18].

3.5 Conclusions

The field of sequence representation learning has seen a multitude of different results over the recent years, which indicates the importance of this field in modern applications. The literature on this subject proposes a multitude of ideas in different domains, including the domain of transactional data

Despite the large amount of work already accomplished in this field, there is a definite lack of models, which perform good both on global and local tasks. Moreover, there still is no common validation procedure for transactional data, which jointly validates the local, global and dynamic properties of these models. This work aims to solve this novel task, and rank the existing approaches accordingly.

Another important contribution of this work lies in taking the external information into account when building representations.

The results of this work will help improve the modern solutions of real-world problems, by incorporating the ideas this work contributes and further deepening our understanding of transaction sequence representations.

4 Methods

4.1 Methods for constructing representations of transactional data

This section describes the methods used to solve problems in the project. We begin this part with a description of specific models from baseline approach: contrastive method CoLES in section 4.1.1. Next, we describe the approaches used to take into account external representations in section 4.2. We finish this section with methods for validating the resulting embeddings in terms of global, local, and dynamic properties.

4.1.1 Contrastive models for transaction sequences

As contrasting methods for self-supervised learning in this work, we chose modern approach CoLES, which shows good results working with event sequence data.

CoLES method for transaction sequences CoLES was used as a starting point in the study of methods for obtaining representations of transactional data. It is a contrastive approach to self-supervised learning, proposed in the work [2]. CoLES shows high quality on a number of tasks of event sequence processing, including transactional data. A comprehensive description of this method can be found in the original article. Below you can find an overview of this approach. We mainly focus on its features that can be important in the study dedicated to the local and global properties of event sequence representations.

CoLES (Contrastive Learning for Event Sequences with Self-Supervision) is an approach for obtaining representations of event sequences, including bank customer transactions, without using target data. This method works in the Self-Supervised Learning paradigm and belongs to the class of contrastive approaches [19].

In the case of CoLES, subsets of one bank client's transactions are used as positive pairs, and subsets of transactions obtained from different clients are used as negative pairs. A parametric encoder model is used to obtain representations of these subsequences. This encoder is trained using a special loss function.

The authors of the method provide various ways to obtain positive and negative pairs. However, they stop at the option in which two transaction subsequences of random size obtained from the users' history are used to obtain a positive pair. In this case, transactions are not mixed in terms of time and their temporary structure is preserved. Negative pairs are made up of the same subsequences but from different clients. In this work we use the same approach.

CoLES model structure The CoLES concept allows to use different encoders to obtain transaction representations. This opportunity is supported by the library pytorch-lifestream, which we used to implement the methods in the current project. In this package, the model structure includes a number of common elements, described below: feature space, transaction encoder, sequential data processing model.

Feature space for constructing representations. In the original work to study the quality of representations from CoLES, the authors considered different sets of features for different datasets. In this project, we standardize the approach and focus on two main characteristics that are present in all datasets: the MCC code of the transaction and its volume in terms of money — Amount. This solution will not only make it possible to more accurately compare different models, but also unify the process of testing the resulting representations.

Transaction encoder (TrxEncoder). Before a sequence of transactions is put into a parametric model, it is processed using a special transaction encoder. The basic procedure involves obtaining Word2Vec-style representations of MCC codes of some fixed dimension d_{mcc}, where each transaction type is associated with a [20] vector. Simple preprocessing of numerical characteristics is also performed, for example, normalization of the Amount.

Thus, TrxEncoder produces a sequence of dimension $(T, d_{mcc} + N_{num})$, where N_{num} is the number of numerical features ($N_{num} = 1$ in the standard case), and T is the length of the transactions sequence.

Model of sequential data processing (SeqEncoder). The resulting sequence of encoded transactions is fed to the main model, which determines the structure of the considered representations. Any neural network that

operates in Sequence-to-Sequence mode can be used as a SeqEncoder. In other words, when the model receives a certain sequence of observations as input, it produces a sequence of representations of the same length. Originally the article proposed to use the recurrent neural networks with the Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) architecture depending on the given dataset.

Limitations of the original approach As already mentioned, the limitations of CoLES are strongly connected to the method, which is used to obtain positive and negative pairs of subsequences. This algorithm explicitly encourages the model to ensure that the resulting representations show information about the user as a whole, rather than about his local state at some time point.

In addition, in the article, the authors use classification of users as the main task to show the quality of their model. This task also shows only global, but not local properties of representations.

This work aims to study not only global but also local properties of transactional data representations. So, the main goal of the proposed improvements to CoLES and other proposed models is to overcome this limitation.

4.2 Usage of external information based on local representations of transactional data

Accounting of external information is a separate important task of this project. We decided to use other users' representations to add context information to the local embeddings of the considered client. In this section, we describe in detail the methodology for solving this problem.

4.2.1 General pipeline for aggregation of external information

An additional context representation vector (we also call it global embedding) is built from the local representation vectors from all or some selected users by aggregating them in various ways.

The global embedding model is built on top of the transaction sequence encoder. Both a pre-trained encoder from the CoLES model and encoders from other models discussed in this work, for example, from the autoencoder, can be used.

The procedure for constructing a context vector at a certain point in time is presented in Figure 1 and is described as follows:

1. We collect a sample of all possible local customer representations: for all users and for each unique moment of the transaction.

2. All local representations from the dataset close to the current time point, but before it are selected.

3. Aggregation is applied to the resulting set of vectors. The resulting vector is the vector of the contextual global representation.

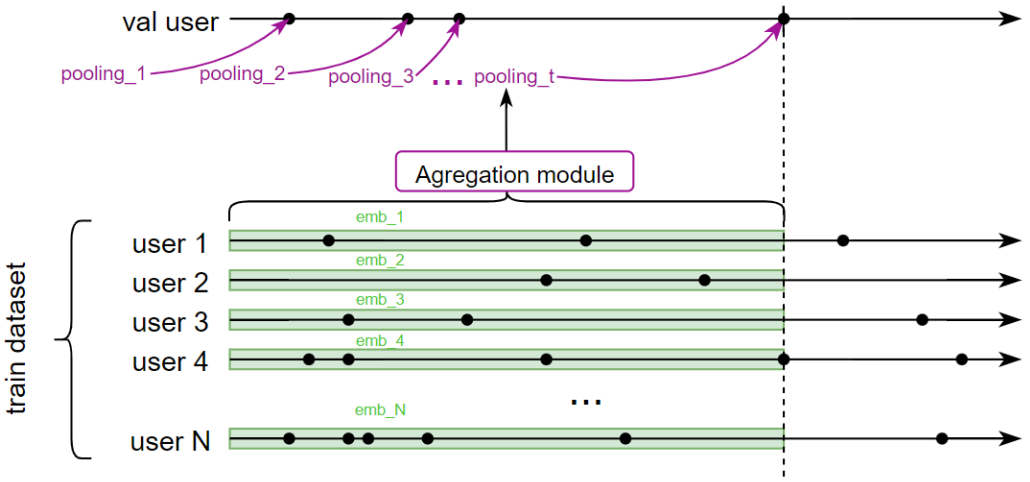


Figure 1: General pipeline for obtaining global representation vectors.

As mentioned earlier, global context vector accounting can improve the quality of models in applied problems. To check this, we concatenate the resulting global embedding vector with the user's local embedding and validate this extended representation.

4.2.2 Classical methods of context vector aggregation

Averaging and maximization were used as classical aggregation methods. In the first case, the vector of global representation is obtained by componentwise averaging of the local representation vectors for all users in the

stored dataset. In the second case, by taking the maximum value for each component. For these methods, we can draw an analogy to the Mean and Max Pooling operations in convolutional neural networks [21]. Just like in convolutional networks, such aggregation methods are designed to generalize the environment for the following manipulation.

4.2.3 Methods based on the attention mechanism for global representations

The problem with the aggregation methods described in the previous paragraph may be the fact that they do not interact with the local representation of the considered user.

Some users in the dataset may behave more similar to him than others. Similar clients determine the user's closest environment and, correspondingly, help to better describe his behavior.

Thus, we need an aggregation method that can explicitly take into account the similarity of users and, respectively, their local representations. The attention mechanism, which was originally used to describe the similarity of the word embedding vectors in different languages in the machine translation task [22], is ideal for such a problem.

In this work, two variants of the attention mechanism are used: without a learnable attention matrix and with it.

In the version without a training matrix, the global context vector for a given point in time has the form:

$$\mathbf{B}_t = Xsoftmax(X^T\mathbf{h}_t), \tag{1}$$

where $\mathbf{h}_t \in R^m$ is a vector of local representation for the considered user, and $X \in R^{m\times n}$ is a matrix, which rows are embeddings of all n users at a given time.

For the method with a trained matrix, the formula is similar:

$$\mathbf{B}_t = Xsoftmax(X^T A\mathbf{h}_t), \tag{2}$$

here $A \in R^{m\times m}$ – is the matrix to be trained.

In the first case, the user similarity metric is normalized by the softmax of the scalar product. In the second, before calculating the scalar product, vectors of representations from the dataset are additionally passed through a trained linear layer. Training of the attention matrix is built into the CoLES model training pipeline.

4.3 Approaches to measure the quality of received representations

The main goal of this work is to study the local and global properties of transactional data representations obtained using various neural network methods.

From our point of view, global properties characterize the behavior of the client as a whole, throughout the entire history of his transactions. In contrast, local properties show the nature of the client's current state at a particular point in time. Moreover, different clients may be similar to each other at some point in time, in which case their local representations should also be similar. In addition, the local properties of the same client can change over time (dynamic property) and the neural network must respond to this change.

Procedures for measuring the quality of representations in terms of their global and local properties are described below.

4.3.1 Methods for validation of the global property

To assess the quality of the obtained representations in terms of global properties, we followed the approach used in the paper [2], in which the CoLES model was proposed.

We used the Churn and Default datasets (see 4.5.1) to solve the problem of binary classification of users on clients who leave the bank and clients who did not repay the loan, respectively. This procedure consists of three steps, which are described below.

For an initial sequence of transactions of length T_i related to the ith user, we obtain the representation $\mathbf{H}_i \in R^d$, which characterizes the entire sequence of transactions as a whole. In the case of using sequence-to-sequence architectures, the max pooling operation is used to obtain a single representation of the entire series from a sequence of views.

Given fixed representations $\mathbf{H}_i$, we predict the binary target label $y_i \in \{0, 1\}$ using gradient boosting. As a specific implementation, the LightGBM [23] model is used, which works fast enough for large data samples and allows obtaining results of sufficiently high quality. The specific hyperparameters of the gradient boosting model are fixed (corresponding to [2]) and are the same for all base models under study.

The quality of the solution of a binary classification problem is measured using a standard set of metrics described in section 4.4.

This procedure allows you to evaluate how well the representations capture the client's "global" pattern across its history.

4.3.2 Local validation methods

Measurement of the quality of the obtained representations in terms of local properties is studied in various formulations.

In all cases, a sliding window procedure of size w is used to obtain local representations. To do this, for the ith user at time $t_j \in [t_w, T_i]$ the subsequence of his transactions $\mathbf{S}^i_{j-w:j}$. Next, this interval is passed through the encoder model under consideration to obtain a local representation $\mathbf{g}^i_j \in R^d$. An illustration of this approach can be found in Figure 2.

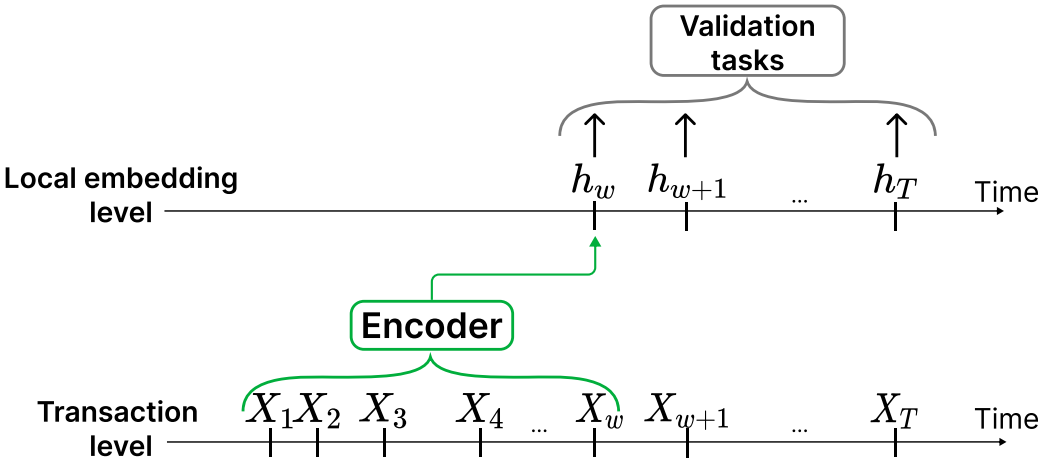


Figure 2: Usage of the sliding window for local validation approaches

The longer time interval the model uses, the greater the risk that the data it relies on will become outdated. Our artificial limitation allows us to reduce this effect for all models and also make their local properties stronger only due to the "relevance" of the data.

An obvious limitation of this approach is that it does not allow obtaining local representations at times $t \leq t^i_w$. In addition, the window size w is an additional hyperparameter that must be chosen, taking into account the fact that for small values of w the resulting representations do not contain

enough information to solve local problems, and for large values of w the representations lose local properties and become global in the limit $w \to T_i$.

As local problems, we explored different approaches, ranging from solving applied problems to studying the dynamics of representations. Below is a detailed description of each method.

Applied problems of transaction classification. For the Churn and Default samples studied in the project (see description in section 4.5.1), it is proposed to consider the corresponding local binary classification tasks (downstream tasks).

In the Churn dataset, target (global) labels correspond to those clients who stopped using the bank's services at some time horizon. It is logical to assume that the behavior of customers who are about to stop using a bank card begins to change in advance: they make fewer transactions, do not top up the card, transactions become less frequent, etc.

Given this, we created local binary labels $c_j^i \in \{0, 1\}$ for all transactions in the sequence. For empirical reasons, it was proposed to select the "early outflow" horizon equal to one month. That is, all transactions of a user who left the bank during the month before his last transaction were marked with the 1 tag, and the rest were marked with the 0 tag.

For the Default sample, similar local binary labels were created, which reflected the client's transition to the "pre-default" state.

Thus, to measure the local properties of embeddings, the problems of their binary classification were considered: the label c_j^i was predicted from the local representation $\mathbf{h}_j^i$. This problem is solved using a simple neural network model; in particular, in this work, we use a single-layer perceptron. Accordingly, the better the embedding describes the behavior of the client "at the moment", that is, locally, the better a simple classifier model will be able to solve applied problems.

Predict the MCC code of the next transaction. This validation approach was inspired by the work [9], in which it was proposed to predict the type of the next event based on the history of observations — in our case, the MCC code of the next transaction.

Formally, in this case, the multiclass classification problem is solved: using the local representation $\mathbf{h}_j^i$, the MCC code of the transaction of the ith client, completed at time t_{j+1}, was predicted.

Note that there are a lot of types that are rare in datasets. From a business perspective, such categories are often less interesting and meaningful. Therefore, to simplify the task, in this procedure for testing local properties,

it was decided to leave only transactions that correspond to the 100 most popular codes. For more information on the distribution of transaction types, see section 4.5.1.

4.4 Metrics

This section presents metrics used to check the quality of work of various models on the above set of tasks.

4.4.1 Classification metrics

Accuracy and F1-Score.

For binary classification problems, we consider the standard metrics Accuracy and F1-Score. For multi-class classification problems, micro-averaging was used for Accuracy and F1-Score. With this averaging method, the metrics are insensitive to class imbalance, since by construction the contribution of each class is proportional to the number of its representatives.

ROC-AUC and PR-AUC.

Also, ROC-AUC and PR-AUC metrics were used for binary and multiclass classification problems. Weighted averaging was used for multiclass classification. This averaging method is also insensitive to class imbalance.

4.5 Datasets

This section describes the datasets used to compare the models and methods under study.

4.5.1 General overview of the data. EDA

In this work, we work with two open samples of transactional data: Churn and Default . Descriptions of these datasets are given below. The main characteristics of the samples are given in Table 1.

Churn. This dataset contains transactional data of bank customers. For global validation (see 4.3), it is proposed to solve the problem of binary classification of clients depending on whether the client left the bank or not. At the

[0]https://boosters.pro/championship/rosbank1

[0]https://boosters.pro/championship/alfabattle2

same time, the classes are almost balanced. The data was used previously, for example, in the work [2].

Default. The dataset also contains transactional data of bank customers. For global validation, it is proposed to solve the problem of binary classification of clients depending on whether the client was able to repay the loan to the bank. There is a significant class imbalance in this sample. This selection was offered to participants at one of the competitions of a large bank and is available in the repository `pytorch-lifestream`. Unlike most other open transaction data in this dataset, the target variable is close to real business problems, as it corresponds to a credit scoring problem.

Table 1: Basic statistics of dataset used in experiments

	Churn	Default
Number of transactions	490 513	2 124 000
Number of clients	5 000	7 080
Min. length of sequences	1	300
Max. length of sequences	784	300
Median length of sequences	83	300
Number of MCC codes	344	309
Class Balance	0.55 : 0.45	0.04 : 0.96

4.5.2 Data preprocessing

For more convenient work with data, we carried out minimal preprocessing. We brought all-time data to one format, removed unnecessary columns, and encoded MCC codes by frequency (i.e. 1 is the most frequent MCC code, 345 is the rarest). The last change is done so that there are no gaps in the categorical task, and to make it technically easier to take into account rare MCC codes (as, for example, in problems with generative models).

To solve the local problem, it was also necessary to add local targets: "will the client leave within a month" in the case of Churn, and "will the client go bankrupt within a month" in the case of Default. These labels are generated using the original target labels available in the datasets.

4.5.3 Conclusions

In this project, we consider specific for the banking industry transactional data. We will conduct experiments on two datasets, Churn and Default.

The presented datasets have sufficiently large sizes and high quality. This ensures the reliability of statistical conclusions and also reduces the risk of overtraining the neural network models used in the project with a large number of parameters. With a high probability, the obtained results will be reproduced on larger datasets that are used to build models in the bank.

In addition, these data have been used previously in other studies. This allows us to correctly compare the resulting models with those previously built for these samples.

Thus, the selected datasets allow us to fully evaluate the quality of the developed methods. Since they are large and have been used previously, the conclusions drawn from these data will accurately reflect the effectiveness and applicability of the proposed methods in real-world settings.

5 Results and Discussion

Model for contrastive learning CoLES: have recurent neural network LSTM with hidden size equal to 1024 for Churn and 800 for Default, batch size equal to 128, maximum number of epoches equal to 60.

For global validation we follow the procedure described in [2] and use the same hyperparameters for boosting model.

For local validation we follow the procedure described in section 4.3.2 and use the following hyperparameters: window size equal to 32, stride equal to 16, batch size equal to 512 and maximum number of epoches equal to 10.

5.1 Experiments using external information

In this section, we describe the results of experiments on obtaining an external context representation and using it to improve existing models.

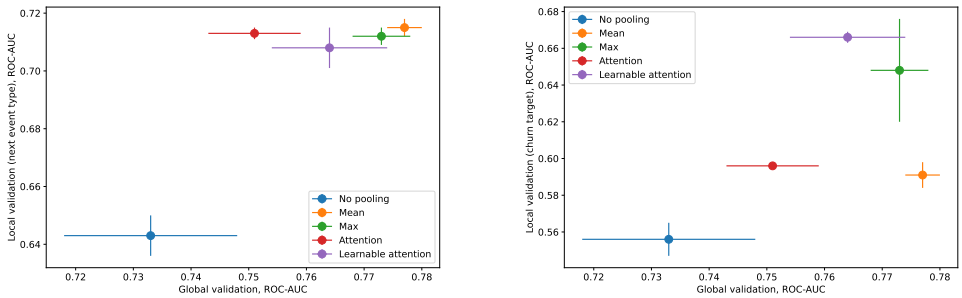
A model using external information requires quite a lot of computing resources, since it needs to store local representations of all users from the training set for all time points, days (see 4.2). To obtain these local representations, we use the encoder from the CoLES trained model.

Due to available memory requirements, we store only a random part of the set of local representations: for the Churn dataset, the number of clients to train in all experiments was 500, and for the Default dataset, 150.

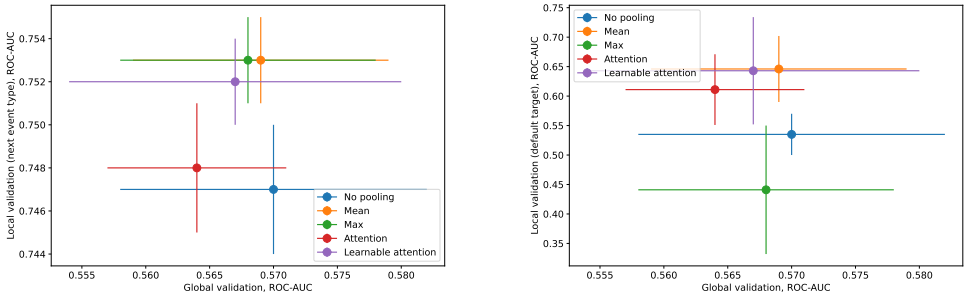
In the experiments, we investigated the following types of aggregation of representations from to obtain a context vector: averaging (Mean), maximization (Max), attention mechanism without a learning matrix (Attention) and with a learning matrix (Learnable attention). The results were compared with a conventional CoLES encoder without adding global information (No pooling). The Learnable attention matrix is trained as part of CoLES contrastive learning. In this case, all weights, except the matrix itself, are fixed and are not trained.

A smaller batch size was also used — 8. The remaining hyperparameters for the encoder remained the same as in experiments with the regular CoLES model.

Models using external information were tested in the same way as other models for obtaining representations of transactional data (see 4.3.2). The experimental results for the Churn and Default datasets are presented in the figure 3 and in the tables 2 and 3. All results were averaged across different pretrained encoder models.



(a) Results for Churn sample



(b) Results for the Default sample

Figure 3: Comparison of the quality of various different types of aggregation in the global representation model on local and global validation. As a local problem, we consider predicting the next type of event (on the left) and the applied problem of classifying transactions (on the right)

5.2 Conclusions from experiments with the model of global representations

Experimental results show that using global representations improves metrics in most cases. This is especially noticeable in the Churn sample, for which the metrics improve across all testing procedures. On the Default sample, contextual representations also help models solve local problems, but according to global validation methods, the results are comparable to the original approach.

Among all the above approaches, the Learnable attention method can be distinguished: it most often ends up among the leaders or on a par with them,

Table 2: Results of different types of validation for different types of aggregation in the global view model on the Churn dataset

	F1-Score ↑	ROC-AUC ↑	PR-AUC ↑
	Global Validation		
No pooling	0.740 ± 0.012	0.733 ± 0.015	0.772 ± 0.026
Mean	0.7527 ± 0.005	**0.777 ± 0.003**	**0.827 ± 0.003**
Max	**0.7534 ± 0.003**	0.773 ± 0.005	0.822 ± 0.007
Attention	0.7401 ± 0.008	0.751 ± 0.008	0.798 ± 0.009
Learnable attention	0.7497 ± 0.004	0.764 ± 0.010	0.815 ± 0.001
	Local validation. Downstream task		
No pooling	0.000 ± 0.000	0.556 ± 0.009	0.316 ± 0.014
Mean	0.000 ± 0.000	0.591 ± 0.007	0.345 ± 0.013
Max	0.000 ± 0.000	0.648 ± 0.028	0.402 ± 0.023
Attention	0.000 ± 0.000	0.596 ± 0.001	0.341 ± 0.014
Learnable attention	**0.097 ± 0.059**	**0.666 ± 0.003**	**0.427 ± 0.006**
	Local validation. Next event type		
No pooling	0.232 ± 0.002	0.643 ± 0.007	0.162 ± 0.005
Mean	0.262 ± 0.006	**0.715 ± 0.003**	**0.214 ± 0.001**
Max	0.261 ± 0.005	0.712 ± 0.003	0.210 ± 0.002
Attention	**0.274 ± 0.007**	0.713 ± 0.002	0.211 ± 0.002
Learnable attention	0.264 ± 0.006	0.708 ± 0.007	0.211 ± 0.003

especially in local validation tasks. This is not surprising since approaches based on the attention mechanism help to well identify local patterns in sequences [22].

In global validation tasks, the Mean method is most often the best; the Max method is also a strong option. This can be explained as follows: these two methods provide some average representation of all users, which can help in global classification tasks, but because of this they can also smooth out the local representations of individual users, which negatively affects the quality of local tasks.

Table 3: Results of different types of validation for different types of aggregation in the global view model on the Default dataset

	F1-Score ↑	ROC-AUC ↑	PR-AUC ↑
	Global Validation		
No pooling	0.000 ± 0.000	**0.570 ± 0.012**	0.058 ± 0.010
Mean	0.000 ± 0.000	0.569 ± 0.010	**0.063 ± 0.010**
Max	0.000 ± 0.000	0.568 ± 0.010	0.061 ± 0.011
Attention	0.000 ± 0.000	0.564 ± 0.007	0.060 ± 0.008
Learnable attention	0.000 ± 0.000	0.567 ± 0.013	0.062 ± 0.008
	Local validation. Downstream task		
No pooling	0.000 ± 0.000	0.535 ± 0.035	0.006 ± 0.000
Mean	0.000 ± 0.000	**0.646 ± 0.056**	**0.010 ± 0.003**
Max	0.000 ± 0.000	0.441 ± 0.109	0.005 ± 0.001
Attention	0.000 ± 0.000	0.611 ± 0.060	0.009 ± 0.003
Learnable attention	0.000 ± 0.000	0.643 ± 0.091	0.009 ± 0.004
	Local validation. Next event type		
No pooling	0.335 ± 0.005	0.747 ± 0.003	0.262 ± 0.002
Mean	**0.340 ± 0.006**	**0.753 ± 0.002**	**0.267 ± 0.002**
Max	0.338 ± 0.006	**0.753 ± 0.002**	**0.267 ± 0.001**
Attention	0.337 ± 0.007	0.748 ± 0.003	0.263 ± 0.001
Learnable attention	**0.340 ± 0.006**	0.752 ± 0.002	**0.267 ± 0.002**

6 Conclusion

In this work, we consider the the task of creation representations for transaction sequences, which are non-uniform event sequences. We adapted existing model CoLES based on contrastive approach. And additionally, we introduced novel methods for incorporating external contextual information, thereby enhancing the quality of existing approaches and foundational solutions.

Also one of the main goal for the project was the creation of the validation which can show both local, and global properties of representations.

The conducted experiments show that the developed approaches naturally addressed project objective: considering external information from other clients' transactions improve the quality in all validation tasks. The most effective results were achieved using an approach based on a trainable attention mechanism, which identifies the closest clients influencing the user under consideration. Mean and Max aggregations also shows outperforming results in global task.

References

[1] Bin Sulaiman Rejwan, Schetinin Vitaly, Sant Paul. Review of machine learning approach on credit card fraud detection // Human-Centric Intelligent Systems. — 2022. — Vol. 2, no. 1-2. — P. 55–68.

[2] Babaev D. et al. CoLES: Contrastive Learning for Event Sequences with Self-Supervision // Proceedings of the 2022 International Conference on Management of Data. — 2022.

[3] Li Tie, Kou Gang, Peng Yi. A new representation learning approach for credit data analysis // Information Sciences. — 2023. — Vol. 627. — P. 115–131.

[4] Learning latent representations of bank customers with the variational autoencoder / Mancisidor Rogelio A, Kampffmeyer Michael, Aas Kjersti, and Jenssen Robert // Expert Systems with Applications. — 2021. — Vol. 164. — P. 114020.

[5] Romanenkova E. et al. Similarity learning for wells based on logging data // Journal of Petroleum Science and Engineering. — 2022.

[6] Ts2vec: Towards universal representation of time series / Yue Zhihan, Wang Yujing, Duan Juanyong, Yang Tianmeng, Huang Congrui, Tong Yunhai, and Xu Bixiong // Proceedings of the AAAI Conference on Artificial Intelligence. — 2022. — Vol. 36. — P. 8980–8987.

[7] Begicheva M., Zaytsev A. Bank transactions embeddings help to uncover current macroeconomics // 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). — 2021.

[8] A deep learning model for behavioural credit scoring in banks / Ala'raj Maher, Abbod Maysam F, Majdalawieh Munir, and Jum'a Luay // Neural Computing and Applications. — 2022. — P. 1–28.

[9] Zhuzhel V. et al. Continuous-time convolutions model of event sequences // arXiv preprint arXiv:2302.06247. — 2023.

[10] Deldari S. et al. Time series change point detection with self-supervised contrastive predictive coding // Proceedings of the Web Conference 2021. — 2021.

[11] A survey on contrastive self-supervised learning / Jaiswal Ashish, Babu Ashwin Ramesh, Zadeh Mohammad Zaki, Banerjee Debapriya, and Makedon Fillia // Technologies. — 2020. — Vol. 9, no. 1. — P. 2.

[12] Self-attentive Hawkes process / Zhang Qiang, Lipani Aldo, Kirnap Omer, and Yilmaz Emine // International conference on machine learning / PMLR. — 2020. — P. 11183–11193.

[13] Hoffer Elad, Ailon Nir. Deep metric learning using triplet network // Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3 / Springer. — 2015. — P. 84–92.

[14] A simple framework for contrastive learning of visual representations / Chen Ting, Kornblith Simon, Norouzi Mohammad, and Hinton Geoffrey // International conference on machine learning / PMLR. — 2020. — P. 1597–1607.

[15] Emerging properties in self-supervised vision transformers / Caron Mathilde, Touvron Hugo, Misra Ishan, Jégou Hervé, Mairal Julien, Bojanowski Piotr, and Joulin Armand // Proceedings of the IEEE/CVF international conference on computer vision. — 2021. — P. 9650–9660.

[16] Zbontar J. et al. Barlow twins: Self-supervised learning via redundancy reduction // International Conference on Machine Learning / PMLR. — 2021.

[17] Thomas Lyn C. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers // International journal of forecasting. — 2000. — Vol. 16, no. 2. — P. 149–172.

[18] Shchur Oleksandr, Biloš Marin, Günnemann Stephan. Intensity-Free Learning of Temporal Point Processes // International Conference on Learning Representations. — 2019.

[19] Self-Supervised Learning: Generative or Contrastive / Liu Xiao, Zhang Fanjin, Hou Zhenyu, Mian Li, Wang Zhaoyu, Zhang Jing, and Tang Jie // IEEE Transactions on Knowledge and Data Engineering. — 2023. — Vol. 35, no. 1. — P. 857–876.

[20] Pennington Jeffrey, Socher Richard, Manning Christopher D. Glove: Global vectors for word representation // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). — 2014. — P. 1532–1543.

[21] Boureau Y-Lan, Ponce Jean, LeCun Yann. A theoretical analysis of feature pooling in visual recognition // Proceedings of the 27th international conference on machine learning (ICML-10). — 2010. — P. 111–118.

[22] Attention is all you need / Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, Kaiser Łukasz, and Polosukhin Illia // Advances in neural information processing systems. — 2017. — Vol. 30.

[23] Lightgbm: A highly efficient gradient boosting decision tree / Ke Guolin, Meng Qi, Finley Thomas, Wang Taifeng, Chen Wei, Ma Weidong, Ye Qiwei, and Liu Tie-Yan // Advances in neural information processing systems. — 2017. — Vol. 30. — P. 3146–3154.