Online Credit Payment Fraud Detection via Structure-Aware Hierarchical Recurrent Neural Network

Wangli Lin[1] , **Li Sun**[1] , **Qiwei Zhong**[1] , **Can Liu**[1] , **Jinghua Feng**[1*] , **Xiang Ao**[2*] , **Hao Yang**[1]
[1]Alibaba Group, Hangzhou, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{wangli.lwl, li.sunl, yunwei.zqw, yiyong.lc, jinghua.fengjh, youhiroshi.yangh}@alibaba-inc.com,
aoxiang@ict.ac.cn

Abstract

Online credit payment fraud detection plays a critical role in financial institutions due to the growing volume of fraudulent transactions. Recently, researchers have shown an increased interest in capturing users' dynamic and evolving fraudulent tendencies from their behavior sequences. However, most existing methodologies for sequential modeling overlook the intrinsic structure information of web pages. In this paper, we adopt multi-scale behavior sequence generated from different granularities of web page structures and propose a model named SAH-RNN to consume the multi-scale behavior sequence for online payment fraud detection. The SAH-RNN has stacked RNN layers in which upper layers modeling for compendious behaviors are updated less frequently and receive the summarized representations from lower layers. A dual attention is devised to capture the impacts on both sequential information within the same sequence and structural information among different granularity of web pages. Experimental results on a large-scale real-world transaction dataset from Alibaba show that our proposed model outperforms state-of-the-art approaches. The code is available at https://github.com/WangliLin/SAH-RNN.

1 Introduction

Recent year dramatically increased e-commerce payments have resulted in the booming occurrence of fraudulent transactions. It has been reported that $24.26 billion was lost due to digital payment fraud worldwide in 2018, which increased by 18.4% compared to 2017 and is still climbing[1]. Online credit payment fraud detection is therefore increasingly important to restrain the impact of fraud on the quality of services, costs and reputation of financial service institutions. Machine learning methods based on feature engineering have played an essential role in financial fraud detection [West and Bhattacharya, 2016; Abdallah *et al.*, 2016], which depends on the effectiveness of the statistical features extracted from different aspects, such as user profiles and historical transactions [Bahnsen *et al.*, 2016]. However, criminals may commit few transactions before making a fraudulent payment, or pretend to be a benign user to make multiple small normal payments and then make a large fraudulent payment. Under these scenarios, the statistical-based features may fail to effectively capture these users' fraudulent patterns, which results in a misjudgement [Zhong *et al.*, 2020].

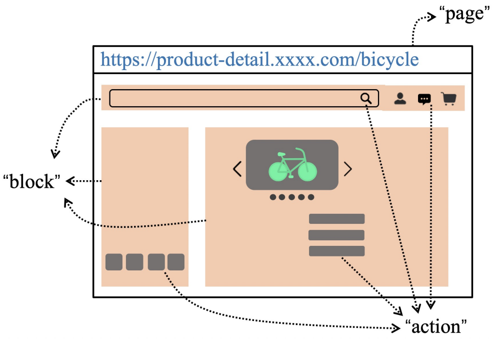

Figure 1: A toy example of web page structure. Each page is composed of several blocks, and users' actions occur under these blocks.

Recently, RNN-based models show promising results by utilizing users' sequential behaviors that can timely reflect their dynamic and evolving intentions [Jurgovsky *et al.*, 2018; Feng *et al.*, 2019]. In these methods, actions of a user on a web page, e.g., click, search, view image, etc., are usually considered as his/her behavior sequence, and the intrinsic structure information of web pages are usually overlooked. A toy example of the intrinsic structure information of web page is demonstrated in Figure 1. In such a figure, a *page* consists of several *blocks*, and each block may contain multiple *actions*. The schema *page*→*block*→*action* is the intrinsic structure of web page used in this paper[2]. Multi-scale behaviors over different granularities can be derived given specific intrinsic web page structures, which might be beneficial to characterize the users' intentions. First, multi-scale behaviors may complement each other. For instance, Figure 2 demonstrates two multi-scale behavior sequences with the same *action* and *block* sequences. However, considering

[*] Corresponding author

[1] https://shiftprocessing.com/credit-card-fraud-statistics/

[2] Other schema is also alternative in our model.

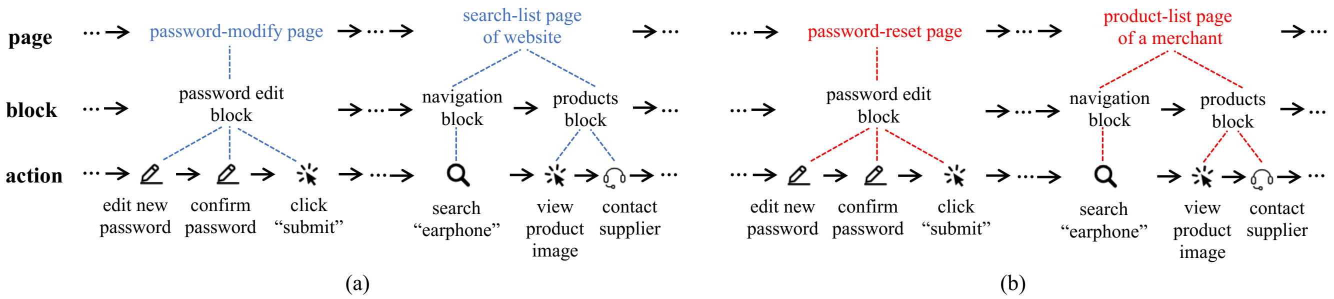

Figure 2: Two examples collected from a real-world dataset. There are many of the same action-level and block-level behaviors while different at page-level in (a) and (b). We can find that the password editing operation in (b) is performed on password reset page whose entry is the password-retrieve page, and the search operation is executed on a particular merchant page, reflecting that the user has a clear purchase intention. Such a pattern is more likely to be suspected as a fraud transaction.

information in the *page* sequence, the potential frauder could be unearthed. Second, the natural tree-like subordinate relations of web page structure shed new light on aggregation multi-scale behaviors to capture hierarchical collaborative behavioral patterns. For example, action sequences in the same block may together be helpful to reflect user's intention in such a block.

To comprehensively exploit multi-scale user behavior sequences derived by web page structures, we propose a novel method coined **S**tructure-**A**ware **H**ierarchical **R**ecurrent **N**eural **N**etworks (**SAH-RNN** for short) in this paper. First, we stack multiple RNN layers to consume multi-scale behavior sequences. Each layer models the corresponding behavior sequence on given granularity of web page structure. Then a subordinate relation guided update and interaction approach among RNN layers is devised. Basically, the upper layers which model compendious behaviors are updated less frequently and receive the summarized representations from lower layers. Furthermore, a dual attention mechanism is designed to perceive the impacts on both sequential information within the same sequence and structural information among different granularity of web pages.

The main contributions are summarized as follows:

- To the best of our knowledge, it is the first work to incorporate web page structures into behavior sequence modeling for online credit payment fraud detection task.
- We propose a novel SAH-RNN which employs stacked RNN layers equipped with the structural subordinate relation guided update strategy and dual attention mechanisms to perceive users' global and local intentions.
- Experimental results on a large-scale real-world dataset from Alibaba platform show that the proposed model significantly outperforms recent state-of-the-art approaches, and has a better generalization ability for various behavior sequence lengths.

2 Related Work

2.1 Credit Payment Fraud Detection

The credit payment fraud occurs when accounts or credit cards are stolen and paid. Conventional solutions usually extract statistical features from different aspects (e.g., user profiles, transaction summarizing and interaction relations) and detect fraudulent transactions with supervised learning algorithms, such as tree-based algorithms [Van Vlasselaer *et al.*, 2015; Xuan *et al.*, 2018] and neural networks [Fu *et al.*, 2016; Liu *et al.*, 2018; Zhang *et al.*, 2018].

However, these methods aggregate historical transactions as a whole and ignore the drifting trend of users' interests. Recent works start to utilize the users' sequential behaviors to enrich the representation of users' dynamic and evolving interests. [Jurgovsky *et al.*, 2018] employs the Long Short-Term Memory Networks (LSTM) to aggregate the users' historical purchase behaviors with the goal to improve fraud detection accuracy. [Babaev *et al.*, 2019] presents Embedding-Transactional Recurrent Neural Network (E.T.-RNN) to compute credit scores of the bank customers by examining their historical credit and debit card transactions. Apart from transaction sequence-based methods, the e-commerce platforms can capture more detailed behaviors in websites (e.g., click, search and edit operations), which have been proven effective for fraud detection in e-commerce scenario. For instance, [Wang *et al.*, 2017] presents a fraud detection system in which the Stacked LSTM is used to capture the users' click and browsing behavior sequences. [Liu *et al.*, 2020] proposes LIC Tree-LSTM that exploits local intention calibration for fraud transaction detection.

Comparatively, both the temporal and hierarchical representations of users' sequential behaviors are modeled in our work, which makes it distinct to the existing ones.

2.2 Hierarchical RNN

Hierarchical structures naturally exist in many sequential data. A typical approach to learn both hierarchical and temporal representation is to stack multiple recurrent layers vertically [El Hihi and Bengio, 1996; Chung *et al.*, 2017], which called hierarchical RNN. Various hierarchical RNNs have been proposed to capture the latent hierarchical structure in sequences. For example, in the clockwork RNN (CW-RNN) [Koutník *et al.*, 2014], the hidden layer is grouped into separate modules, each of which is interconnected and is explicitly assigned different timescales for update. Dilated RNN [Chang *et al.*, 2017] is constructed by stacking multiple dilated recurrent layers with hierarchical dilations, which mitigates the vanishing gradient problem. Ordered

Neurons LSTM (ON-LSTM) [Shen *et al.*, 2019] introduces an elegant way of adding a hierarchical inductive bias to integrate the latent tree structures in sequences into recurrent models, achieving state-of-the-art performance on many sequential tasks.

Inspired by multi-scale behavior sequences that are derived according to the natural tree-like structure of web pages, we propose to incorporate the structural information into hierarchical RNN architecture, which has not been explored before.

3 Methodology

3.1 Problem Statement

In this paper, the online credit payment fraud detection task is phrased as a binary sequence classification problem. That is, given the user's multi-scale behavior sequence $\boldsymbol{X} = [\boldsymbol{X}_1; \boldsymbol{X}_2; ...; \boldsymbol{X}_T]$ before initiating a payment (also known as a transaction), where T is the length of behavior sequence, $\boldsymbol{X}_t \in \mathcal{R}^{L*d}$ denotes the multi-scale behaviors at the t-th time step and L is the number of structure hierarchies, our purpose is to predict whether it is a fraudulent transaction. We assign a label $y \in \{0, 1\}$ on each transaction to indicate whether it is fraud or not. Hence, given the training set $\mathcal{D} = \{(\boldsymbol{X}, y)\}$, our goal is to predict the fraud probability of each transaction in the testing set.

3.2 Model Overview

In the following, we present an SAH-RNN variant based on LSTM, called Structure-Aware Hierarchical LSTM (SAH-LSTM). As shown in Figure 3, the overall architecture has three main components, namely Structure-Aware Factor Sequences Extraction, Hierarchical RNN Layers and Dual Attention Mechanism. The structure-aware factor sequences are extracted from the multi-scale behavior sequences to determine the timing of updating and interacting for the hierarchical RNN layers. The hierarchical RNN consists of multiple stacked RNN layers. Each layer models the corresponding behavior sequence at the given granularity of web page structure and interacts with other layers when the update operation is executed. The dual attention mechanism including sequence attention and structure attention layers are designed to capture the users' global and local intentions simultaneously.

Structure-Aware Factor Sequences Extraction

To automatically capture the structure information of web page, the structure-aware factor sequences $\boldsymbol{S} \in \mathcal{R}^{L*T}$ are extracted from the multi-scale behavior sequences $\boldsymbol{X}$, s_t^l stands for the structure-aware factor in the l-th layer at the t-th time step, which is calculated as follows.

$$s_t^l = \begin{cases} 0 & \text{if } x_t^l = x_{t+1}^l \\ 1 & \text{if } x_t^l \neq x_{t+1}^l \text{ or } t = T \end{cases} \tag{1}$$

The binary value of s_t^l indicates whether the behavior in the next time step varies from the current time step. As the upper layers have more repeated behaviors, the binary sequences $\boldsymbol{S}$ are getting sparse from lower layers to upper layers. Moreover, the structure-aware factor sequences determine whether to update the hierarchical RNN layers and are used as the masking vector of the sequence attention layer.

Hierarchical RNN Layers

Taking the multi-scale behavior sequences and structure-aware factor sequences as input, the depth of the stacked LSTM layers is the same as the number of structure hierarchies, and each layer models a behavior sequence at the corresponding granularity from fine to coarse. At each time step, each layer executes either update operation or copy operation, which is determined by the structure-aware factor s_t^l. The update function for the hidden states $\boldsymbol{h}$ and cell states $\boldsymbol{c}$ at each layer l is defined as follows.

$$\boldsymbol{h}_t^l = \begin{cases} \boldsymbol{h}_{t-1}^l & \text{if } s_t^l = 0 \\ \boldsymbol{o}_t^l \odot \tanh(\boldsymbol{c}_{t-1}^l) & \text{if } s_t^l = 1 \end{cases} \tag{2}$$

$$\boldsymbol{c}_t^l = \begin{cases} \boldsymbol{c}_{t-1}^l & \text{if } s_t^l = 0 \\ \boldsymbol{f}_t^l \odot \boldsymbol{c}_{t-1}^l + \boldsymbol{i}_{t-1}^l \odot \boldsymbol{g}_t^l & \text{if } s_t^l = 1 \end{cases} \tag{3}$$

where $\boldsymbol{f}, \boldsymbol{i}, \boldsymbol{o}, \boldsymbol{g}$ are forget gate, input gate, output gate and candidate vector, respectively, and $\odot$ is the element-wise product. Note that unlike the standard stacked LSTM which are updated at every time step, the upper layers are updated less frequently than the lower layers. It is because the upper layer models the behavior sequence with a coarser granularity. The update operation is executed only when the behavior in the next time step varies from the current, and $\boldsymbol{f}, \boldsymbol{i}, \boldsymbol{o}, \boldsymbol{g}$ are obtained by the following equations.

$$\begin{bmatrix} \boldsymbol{f}_t^l \\ \boldsymbol{i}_t^l \\ \boldsymbol{o}_t^l \\ \boldsymbol{g}_t^l \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} f_{\text{slice}}(\boldsymbol{W}_1^l \boldsymbol{h}_{t-1}^l + \boldsymbol{W}_2^l \tilde{\boldsymbol{x}}_t^l + b^l) \tag{4}$$

where σ is the sigmoid function, $\boldsymbol{W}$ and b denote the weight matrix and bias vector, respectively. $\tilde{\boldsymbol{x}}_t$ is the input to this step, which is defined as follows.

$$\tilde{\boldsymbol{x}}_t^l = \begin{cases} \boldsymbol{x}_t^l & \text{if } l = 1 \\ [\boldsymbol{x}_t^l, \boldsymbol{h}_t^{l-1}] & \text{if } l > 1 \end{cases} \tag{5}$$

At the first layer, the input is the embedding of the finest-grained behavior sequence $\boldsymbol{x}_t^1$. Otherwise, we concatenate $\boldsymbol{x}_t^l$ with the hidden state of the lower layer $\boldsymbol{h}_t^{l-1}$ as the input to the upper layer l. This operation allows the upper layer to absorb the summarized representation of behaviors at the lower layer. Thus, it is reasonable to expect that the upper layer obtains a more comprehensive global view, benefiting from the memorization of long-term dependencies based on short-term dependencies learned from the lower layer.

Sequence Attention Layer

Considering that behaviors at the same LSTM layer are strongly related to each other, and there are many consecutive repeated behaviors exist in the upper layer which are not updated at every step. We employ the multi-head self-attention mechanism [Vaswani *et al.*, 2017] to capture the inner relationship between behaviors at the same layer and decrease the effect of repeated behaviors at the upper layers. Specially, the self-attention operations are applied to the hidden states of each LSTM layer separately. Mathematically, let $\boldsymbol{H}_l = [\boldsymbol{h}_1^l; \boldsymbol{h}_2^l; ...; \boldsymbol{h}_T^l] \in \mathcal{R}^{T*d}$ be the input of the l-th self-attention layer, and the i-th head is calculated as follows.

$$\begin{aligned} \boldsymbol{head}_i^l &= \text{Attention}(\boldsymbol{H}_l \boldsymbol{W}_l^Q, \boldsymbol{H}_l \boldsymbol{W}_l^K, \boldsymbol{H}_l \boldsymbol{W}_l^V) \\ &= \text{softmax}(\frac{\boldsymbol{H}_l \boldsymbol{W}_l^Q \boldsymbol{W}_l^{K^\top} \boldsymbol{H}_l^\top}{\sqrt{d}}) \boldsymbol{H}_l \boldsymbol{W}_l^V \end{aligned} \tag{6}$$

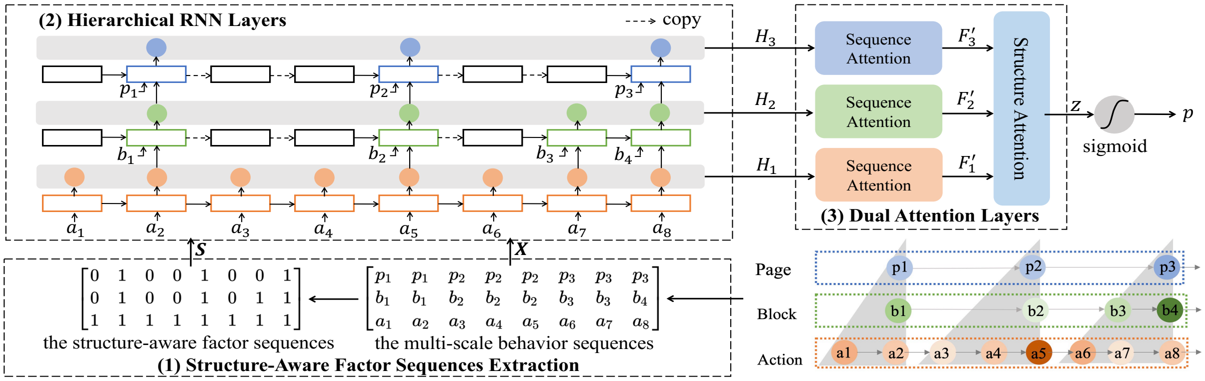
(2) Hierarchical RNN Layers
copy
p_1
p_2
p_3
b_1
b_2
b_3
b_4
a_1
a_2
a_3
a_4
a_5
a_6
a_7
a_8
H_3
H_2
H_1
Sequence Attention
Sequence Attention
Sequence Attention
F_3'
F_2'
F_1'
Structure Attention
z
sigmoid
p
(3) Dual Attention Layers
S
X
0 1 0 0 1 0 0 1
0 1 0 0 1 0 1 1
1 1 1 1 1 1 1 1
the structure-aware factor sequences
p_1 p_1 p_2 p_2 p_2 p_3 p_3 p_3
b_1 b_1 b_2 b_2 b_2 b_3 b_3 b_4
a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8
the multi-scale behavior sequences
(1) Structure-Aware Factor Sequences Extraction
Page
Block
Action
p1
p2
p3
b1
b2
b3
b4
a1
a2
a3
a4
a5
a6
a7
a8

Figure 3: The overall architecture of our proposed model SAH-RNN.

where the projection matrices $\boldsymbol{W}_l^Q, \boldsymbol{W}_l^K, \boldsymbol{W}_l^V \in \mathcal{R}^{d*d}$. As for the input of the softmax function, we use the structure-aware factor sequence at the corresponding layer of S as the masking vector to mitigate the impact of consecutive repeated behaviors on the upper layer. Then the concatenated vector of different heads is fed into the feed-forward network:

$$\boldsymbol{F}_l = \mathrm{FFN}(\mathrm{Concat}(\boldsymbol{head}_1^l, \ldots, \boldsymbol{head}_h^l)\boldsymbol{W}_l^O) \qquad (7)$$

where $\mathrm{FFN}(\cdot)$ is the feed-forward network, $\boldsymbol{W}_l^O \in \mathcal{R}^{hd*d}$ is the projection matrix, and h is the number of heads. Residual connection and layer normalization are successively conducted around both the multi-head attention and feed-forward network. Then the distilled representation of $\boldsymbol{F}_l$ is calculated as follows.

$$\boldsymbol{F}_l' = \mathrm{Avg}\left(\boldsymbol{F}_l\right) \qquad (8)$$

where $\mathrm{Avg}(\cdot)$ is the average pooling.

Structure Attention Layer

We assume that the infrequently updated upper layers have a more global view, while the frequently updated lower layers can better capture short-term intentions. Intuitively, different layers are likely to have different importance for the prediction task. We thus devise a structure attention mechanism to merge all representations of behavior sequences $\boldsymbol{F}'_1, \boldsymbol{F}'_2, ..., \boldsymbol{F}'_L$ into a distilled representation. The definitions of a structure attention mechanism are as follows.

$$e_l = \tanh(\boldsymbol{W_a}^\top \boldsymbol{F}_l' + b_a) \qquad (9)$$

$$a_l = \frac{\exp(e_l)}{\sum_{k=1}^{L} \exp(e_k)} \qquad (10)$$

$$\boldsymbol{z} = \sum_{l=1}^{L} a_l \boldsymbol{F}_l' \qquad (11)$$

where $\boldsymbol{W_a}$ and b_a are the trainable weight matrix and bias, respectively. a_l is the attention weight for the corresponding behavior sequence, and $\boldsymbol{z}$ denotes the final representation of multi-scale behavior sequences generated by SAH-LSTM.

Model Training

The obtained final representation $\boldsymbol{z}$ is then fed into a fully connection layer with a sigmoid unit, and the predicted fraud probability is calculated as follows.

$$p = \sigma(\boldsymbol{w_p}^\top \boldsymbol{z} + b_p) \qquad (12)$$

where $\boldsymbol{w_p}$ and b_p are the weight vector and bias, respectively, and p is the predicted fraud probability. Finally, our model is trained with the negative log-likelihood function, which is defined as follows.

$$\mathcal{L} = -\frac{1}{N} \sum_{(\boldsymbol{X}, y) \in \mathcal{D}} (y \log(p) + (1-y)\log(1-p)) \qquad (13)$$

where $\mathcal{D}$ is the training dataset, the input $\boldsymbol{X}$ is the users' multi-scale behavior sequences, and y is the ground truth.

4 Experiments

4.1 Dataset

To verify the effectiveness of SAH-RNN in the real-world industrial applications, we conduct extensive experiments on a large-scale dataset[3] from Alibaba platform (www.alibaba.com), one of the most popular e-commerce platforms in China. According to the general definition in financial scenarios, we define the positive samples as the fraudulent transactions and the negative samples as genuine transactions. The details of the dataset is summarized in Table 1. For the behaviors over different structure granularities, we did some necessary preprocessing like clustering, and selected top frequent behavior categories as our vocabulary. The detailed statistical information are shown in Table 2.

Dataset	# Positive	# Negative	# Positive Rate
Training	31,216	2,353,543	1.31%
Testing	6,269	4,54,155	1.36%

Table 1: Statistics of datasets.

	Category	Examples
Action	300	edit message/view image/add to cart/…
Block	100	message box/search bar/products block/…
Page	30	homepage/search page/cashier page/…

Table 2: Category statistics of multi-scale behaviors.

[3] only includes website browsing behaviors permitted by the user.

Methods	**AUC**						**R@P$_{0.1}$**					
	All	(0,100]	(100,200]	(200,300]	(300,400]	(400,500]	All	(0,100]	(100,200]	(200,300]	(300,400]	(400,500]
GRU	0.8690	0.6120	0.8272	0.9064	0.9057	0.8916	0.7387	0.4695	0.6311	0.8750	0.8808	0.8038
LSTM	0.8711	0.6672	0.8438	0.9084	0.9139	0.8938	0.7520	0.4695	0.6524	0.8667	0.9025	0.8172
Stacked GRU	0.8728	0.6929	0.8498	0.9001	0.9171	0.8906	0.7510	0.4974	0.6667	0.8625	0.8843	0.8098
Stacked LSTM	0.8766	0.7056	0.8403	0.9029	0.9064	0.8913	0.7625	0.5103	0.6168	0.8583	0.8929	0.8188
TextCNN	0.8801	0.6384	0.8513	0.9099	0.9149	0.9029	0.7588	0.5082	0.6880*	0.8764	0.8964	0.8403
Transformer	0.8872	0.6837	0.8517	0.9067	0.9149	0.9042	0.7885*	0.5146	0.6838	0.8750	0.8964	0.8425*
CW-LSTM	0.8779	0.6730	0.8298	0.9016	0.9156	0.8956	0.7577	0.5082	0.6111	0.8444	0.8912	0.8180
Dilated LSTM	0.8852	0.7210*	0.8549*	0.9135*	0.9210	0.8958	0.7673	0.5275*	0.6795	0.8764*	0.9067	0.8196
ON-LSTM	0.8880*	0.6973	0.8496	0.9129	0.9236*	0.9102*	0.7741	0.5189	0.6481	0.8736	0.9086*	0.8411
SAH-LSTM	**0.9090**	**0.7334**	**0.8820**	**0.9256**	**0.9393**	**0.9266**	**0.8229**	**0.5596**	**0.7179**	**0.8972**	**0.9240**	**0.8843**

Table 3: Performance of different methods. Column "All" indicates the results on the whole testing dataset, and the others are on different testing subsets grouped by behavior sequence length. The * indicates the best performance among the compared methods, and the best results of all methods are indicated in bold face. Five-run-average values on testing set are reported.

4.2 Compared Methods

(1) Sequence-based Methods

- **LSTM** is a widely used RNN variant [Hochreiter and Schmidhuber, 1997]. Here we consider both single-layer LSTM and 3-layer Stacked LSTM.
- **GRU** simplifies the gates in LSTM [Chung *et al.*, 2014]. Both single-layer and 3-layer Stacked GRU are compared.
- **TextCNN** uses multiple one-dimensional convolutional layers with different kernel sizes to capture the dependencies among behaviors [Kim, 2014].
- **Transformer** is an essentially attention based model [Vaswani *et al.*, 2017].

(2) Hierarchical RNN-based Methods

- **CW-LSTM** separates the hidden layer into several modules for processing inputs at different temporal granularity to better capture long-term dependencies [Koutník *et al.*, 2014].
- **Dilated LSTM** is constructed by stacking multiple recurrent layers with multi-resolution dilated recurrent skip connections [Chang *et al.*, 2017]. We consider 3-layer Dilated LSTM for comparison.
- **ON-LSTM** integrates tree structures into recurrent neural networks by ordered neurons to obtain hierarchical representations [Shen *et al.*, 2019]. We consider 3-layer ON-LSTM for comparison.

4.3 Implementation Details

All the models are implemented with Tensorflow [Abadi *et al.*, 2016]. For fair comparisons, we conduct experiments with different configurations on the number of layers and hidden units in various compared models. For each transaction, we backtrack the user's behaviors in the last 7 days before making payment. We limit the maximal length of the behavior sequence to 500 and paddings are performed for too short behaviors. For all the experiments, we under-sampled the negative examples to lift the ratio of positive samples (fraud transactions) at 10% in the training dataset. We randomly extract 10% samples from the original training set for validation, and perform early stopping if the validation performance is not improved for 10 epochs. Moreover, we choose Adam [Kingma and Ba, 2015] as optimizer and decide the initial learning rate from $\{0.01, 0.001, 0.0001\}$ via validation. We set the batch size to 512 and deliberately optimize the parameters in other compared methods according to their literatures.

4.4 Metrics

Since the positive rate in the online payment scenario is really low in general, we evaluate the performance of the approaches with **AUC** and **R@P$_{0.1}$**. The AUC is the area under the ROC curve, which reflects the ranking ability. The R@P$_{0.1}$ is the recall when the precision equals to 10%, which indicates the ability to detect top-ranking positive samples. In summary, the higher both two evaluate metrics are, the better performance the results have.

4.5 Main Results

Table 3 demonstrates the main results. The major findings from the results can be summarized as follows:

(1) Compared with the single-layer RNN variants, our proposed model facilitates the hierarchical RNN architecture to fully exploit multi-scale behavior sequences, and we can clearly observe that our model significantly outperforms single-layer LSTM or GRU by a large margin, e.g., SAH-LSTM outperforms GRU with about 4.00% increased AUC and 8.42% increased R@P$_{0.1}$. Meanwhile, the usage of different update frequencies at different layers makes our models more advantageous than stacked RNN variants. For instance, SAH-LSTM gets 3.24% higher AUC and 6.04% higher R@P$_{0.1}$ than 3-layer stacked LSTM. Furthermore, SAH-LSTM is also more advanced than the CNN-based and attention-based methods, i.e., TextCNN and Transformer, with 6.41% and 3.44% increased R@P$_{0.1}$, respectively.

(2) The explicit boundary information extracted from the structure of web page enables SAH-LSTM to achieve a better performance than other hierarchical RNN models. CW-LSTM and Dilated LSTM update each layer with a fixed but different rate without considering that non-stationarity is prevalent in behavior sequences. ON-LSTM dynamically adapts the update frequencies for different neurons, and its

performance is better than CW-LSTM and Dilated LSTM, but still worse than SAH-LSTM. These observations demonstrate that our proposed models are more effective than other hierarchical RNN models due to taking advantage of structural information and the dual attention mechanism.

(3) Among these baseline methods, we can find that ON-LSTM performs the best on AUC metric while Transformer is the best on R@P$_{0.1}$ metric. We conjecture their improvements mainly come from the consideration of the global intention. In more detail, Transformer treats the entire sequence as a whole and utilizes the self-attention mechanism to model the interactions within the sequence. ON-LSTM, on the other hand, adopts the high-ranking ordered neurons to encode long-term information. However, without the consideration of structural information among multi-scale behavior sequences, they can hardly maximize the performance when the sequence is short, and it is the key factor that SAH-LSTM can outperform these two methods.

4.6 Ablation Test

We perform the ablation study and the results are shown in Table 4. Specifically, we compare SAH-LSTM with its three variants, namely SAH-LSTM$_{\backslash SeqAtt}$ (without sequence attention), SAH-LSTM$_{\backslash StructAtt}$ (without structure attention) and SAH-LSTM$_{\backslash LayerInter}$ (without layer interaction, i.e., the upper layers will not receive the summarized representation from the lower layers). The results show that both AUC and R@P$_{0.1}$ get worse by removing any of these three designs, which reflects that all three designs are important. The sharply decreased R@P$_{0.1}$ in SAH-LSTM$_{\backslash SeqAtt}$ and SAH-LSTM$_{\backslash StructAtt}$ indicates the effectiveness of the dual attention layers. The SAH-LSTM$_{\backslash LayerInter}$ performs worse than other variants, which illustrates that the information transmission between RNN layers has a more significant impact on detecting fraudulent transactions in our dataset.

Sequence Attention	Structure Attention	Layer Interaction	AUC	R@P$_{0.1}$
×	√	√	0.8981	0.7884
√	×	√	0.8978	0.7895
√	√	×	0.8877	0.7830
√	√	√	0.9090	0.8229

Table 4: Ablation analysis of SAH-LSTM.

4.7 Further Discussion

Impact of Behavior Sequence Length

We divide the testing set into 5 groups by the interval of 100 to analyze the impact of different behavior sequence lengths, as shown in Table 3. We further put the sequences whose length more than 500 as a group separately, and the AUC and R@P$_{0.1}$ curves are shown in Figure 4. The results indicate that all models' performance improve obviously with the increase of behavior sequence length at the beginning. However, the performance of the other compared baseline methods begins to deteriorate as the sequence length further increases, while the curves trend of SAH-LSTM are flatter, and achieve better results on both shorter and longer sequences than other compared models. It gives us an encouraging result that our proposed models can better distinguish the fraud patterns for both new customers with few behaviors and mature customers with more complex behaviors.

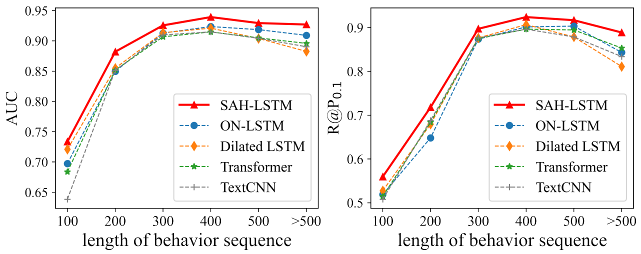

Figure 4: Performance on different sequence lengths.

Effect of Structure Attention Layer

We next visualize the distribution of structure attention weights for the transactions in testing set, as shown in Figure 5. It can be observed that the fraudulent transactions have higher attention weights on action-level behaviors, which are modeled by the frequently updated lower layer and can better capture the local mutational behavior patterns. While the benign transactions pay more attention to page-level behaviors which have a more comprehensive view. We conjecture the possible reason for these phenomena is the structure attention layer can perceive users' global and local intentions.

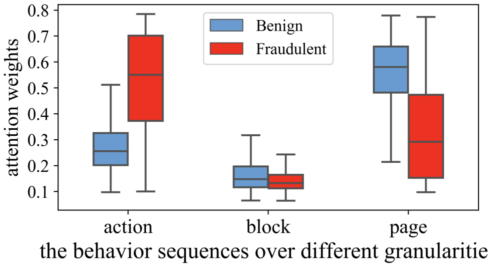

Figure 5: The boxplot of structure attention weights.

5 Conclusion

In this paper, we proposed an SAH-RNN to model multi-scale behavior sequences derived by web page structures. The subordinate relations of web page structures guided the update strategy of the network. A dual attention mechanism was devised to learn interactions among the multi-scale behaviors and simultaneously capture users' global and local intentions. Experimental results on the large-scale real-world dataset demonstrate the effectiveness of the proposed model.

Acknowledgements

This work is supported by Alibaba Innovative Research Program. The work is also supported by the National Natural Science Foundation of China under Grant No.92046003, 61976204, U1836111, the Project of Youth Innovation Promotion Association CAS and Beijing Nova Program Z201100006820062.

References

[Abadi *et al.*, 2016] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016.

[Abdallah *et al.*, 2016] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113, 2016.

[Babaev *et al.*, 2019] Dmitrii Babaev, Maxim Savchenko, Alexander Tuzhilin, and Dmitrii Umerenkov. Et-rnn: Applying deep learning to credit loan applications. In *KDD*, 2019.

[Bahnsen *et al.*, 2016] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, and Björn Ottersten. Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51:134–142, 2016.

[Chang *et al.*, 2017] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. In *NIPS*, 2017.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS*, 2014.

[Chung *et al.*, 2017] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *ICLR*, 2017.

[El Hihi and Bengio, 1996] Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *NIPS*, 1996.

[Feng *et al.*, 2019] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. Deep session interest network for click-through rate prediction. In *IJCAI*, 2019.

[Fu *et al.*, 2016] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. Credit card fraud detection using convolutional neural networks. In *ICONIP*, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Jurgovsky *et al.*, 2018] Johannes Jurgovsky, Michael Granitzer, Konstantin Ziegler, Sylvie Calabretto, Pierre-Edouard Portier, Liyun He-Guelton, and Olivier Caelen. Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100:234–245, 2018.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*. ACL, 2014.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Koutník *et al.*, 2014] Jan Koutník, Klaus Greff, Faustino J. Gomez, and Jürgen Schmidhuber. A clockwork RNN. In *ICML*, 2014.

[Liu *et al.*, 2018] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *CIKM*, 2018.

[Liu *et al.*, 2020] Can Liu, Qiwei Zhong, Xiang Ao, Li Sun, Wangli Lin, Jinghua Feng, Qing He, and Jiayu Tang. Fraud transactions detection via behavior tree with local intention calibration. In *KDD*, 2020.

[Shen *et al.*, 2019] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*, 2019.

[Van Vlasselaer *et al.*, 2015] Véronique Van Vlasselaer, Cristián Bravo, Olivier Caelen, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75:38–48, 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[Wang *et al.*, 2017] Shuhao Wang, Cancheng Liu, Xiang Gao, Hongtao Qu, and Wei Xu. Session-based fraud detection in online e-commerce transactions using recurrent neural networks. In *ECML/PKDD*, 2017.

[West and Bhattacharya, 2016] Jarrod West and Maumita Bhattacharya. Intelligent financial fraud detection: a comprehensive review. *Computers & Security*, 57:47–66, 2016.

[Xuan *et al.*, 2018] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. Random forest for credit card fraud detection. In *ICNSC*, 2018.

[Zhang *et al.*, 2018] Zhaohui Zhang, Xinxin Zhou, Xiaobo Zhang, Lizhi Wang, and Pengwei Wang. A model based on convolutional neural network for online transaction fraud detection. *Security and Communication Networks*, 2018.

[Zhong *et al.*, 2020] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He. Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In *WWW*, 2020.