



Fraud detection via behavioral sequence embedding

Guannan Liu¹ · Jia Guo² · Yuan Zuo¹ · Junjie Wu^{1,3,4} · Ren-yong Guo¹

Received: 12 January 2019 / Revised: 20 December 2019 / Accepted: 27 December 2019 /

Published online: 9 January 2020

© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Fraud detection is usually compared to finding a needle in a haystack and remains a challenging task because fraudulent acts are buried in massive amounts of normal behavior and true intentions may be disguised in a single snapshot. Indeed, fraudulent incidents usually take place in consecutive time steps to gain illegal benefits, which provides unique clues for probing fraudulent behavior by considering a complete behavioral sequence rather than detecting fraud from a snapshot of behavior. Additionally, fraudulent behavior may involve different parties, such that the interaction patterns between sources and targets can help distinguish fraudulent acts from normal behavior. Therefore, in this paper, we model the **attributed behavioral sequences** generated from consecutive behaviors in order to capture the sequential patterns, while those that deviate from the pattern can be detected as fraudulence. Considering the characteristics of the behavioral sequence, we propose a novel model, *NHA-LSTM*, by augmenting the traditional LSTM with a modified forget gate, where the interval time between consecutive time steps is considered. Furthermore, we design a self-historical attention mechanism to allow for long time dependencies, which can help identify repeated or cyclical appearances. In addition, we propose an enhanced network embedding method, FraudWalk, to construct embeddings for the nodes in the interaction network with regard to higher-order interactions and particular time constraints for revealing potential group fraudulence. The node embeddings, along with the feature vectors, are fed into the model to capture the interactions between sources and targets. To validate the effectiveness of sequential behavior embeddings, we experiment on a real-world telecommunication dataset with prediction and classification tasks based on the learned embeddings. The experimental results show that the learned embeddings can better identify fraudulent behavior. Finally, we visualize the weights of the attention mechanism to provide a rational interpretation of human behavioral patterns.

Keywords Fraud detection · Behavioral sequence · LSTM · Network embedding · Attention

Dr. Liu was supported by National Science Foundation of China (NSFC) under Grant 71701007. Dr. Wu was supported by the National Key R&D Program of China (2019YFB2101804) and NSFC under Grants 71725002, 71531001, U1636210, 71490723. Dr. Zuo was supported by NSFC under Grant 71901012 and China Postdoctoral Science Foundation 2018M640045.

✉ Yuan Zuo
zuoyuan@buaa.edu.cn

Extended author information available on the last page of the article

1 Introduction

Fraudulence can be regarded as a type of anomalous behavior with illegal benefits. To disguise their illegal purposes, fraudsters may pretend to act like normal users. Therefore, detecting fraudulent behavior always remains a challenging task due to the relatively scarce appearance of such behavior buried in large amounts of normal behavior, particularly when only one snapshot of behavior is considered. However, some fraudulent behavior is manifested in consecutive time periods since most fraudsters would not succeed through a single contact with the potential victim. In other words, when consecutive behaviors are ordered in a *behavioral sequence*, we would see more potentially anomalous behavioral patterns. For example, when conducting telecom frauds, fraudulent callers may strategically launch consecutive calling behavior toward potential victims to make their fake stories appear more authentic and to induce potential targets to transfer money into their accounts. The behavior at each time step in the behavioral sequence not only is a binary action but also contains multidimensional attributes, which can be referred to as an *attributed behavioral sequence*. For instance, a calling behavior consists of attributes such as the time of the call, target contact, and zone, with each revealing the potential behavioral patterns from a particular perspective. Additionally, sequential patterns such as periodicity, that is, the repeated occurrences of similar behaviors, can also help distinguish fraud from normal behavior because fraudulent behavior may be more random without obvious temporal regularities.

Except for the independent attributes to describe a behavior in the sequences, the interaction structure formed in the behavioral sequences, i.e., whom the users interact with, can also help to probe the normality of their behaviors. In fact, fraudulent behavior usually involves two parties, namely sources and targets, of which the parties who launch fraudulent behaviors are regarded as sources and the potential victims of these fraudulent acts are the targets. In telecommunication scenarios, callers act as sources to illegally induce potential target callees to transfer money. Thus, the interactions between sources and targets are deemed important in distinguishing normal and fraudulent behaviors. For example, normal callers generally interact with a focused group of callees, including their families, colleagues and friends, while fraudulent callers have to dial more diverse callees in a wider range in order to harvest more potential victims.

The interactions between sources who launch the actions and targets who would be affected by the actions can form a dynamic bipartite network; hence, interaction patterns can naturally be manifested in the network structure. In reality, several fraudsters may form a gang to launch their malicious behaviors simultaneously, which would show the phenomenon that a group of source users consecutively targets a focal group of users within a short time period. For instance, in telecom fraud scenarios, a group of fraudsters may play different roles such as police, tax bureaus, and lawyers, and they work together to target the same fraudulent incident to induce the target users to trust them. Therefore, the interaction structure is deemed to play an important role in distinguishing fraudulent and normal behaviors and is indeed formed in the behavioral sequences. However, such interaction structures along with the sequential behaviors are merely addressed simultaneously for the task of fraud detection.

To address the above challenges in fraud detection, we aim to model the behavioral sequences instead of detecting fraudulence from snapshots of behavior. Then, the hypothesis is that by capturing the normal behavioral patterns from the attributed behavioral sequences and the interaction structure simultaneously, behavior that deviates from the normal behavioral patterns can be regarded as potential fraudulence. Indeed, several prior works have attempted to model sequences and map them into low-dimensional space. However, these

studies still suffer from several limitations when taking the goal of fraud detection into account. Specifically, prior sequence learning models, e.g., LSTM and its variations, generally assume fixed intervals between consecutive time steps and relatively short time dependencies, while the interactions across different parties are not captured, which cannot accurately represent the behavioral sequences to identify frauds. In addition, the interactions between sources and targets are generally not modeled to represent the behavior sequence, which limits the discrimination power for the fraudulent behaviors.

Therefore, in this paper, we propose to detect fraudulent behaviors from learning the representations of the attributed behavioral sequences. Basically, behavioral sequences are modeled with an LSTM model, and the attributes at each time step are also fed into the original LSTM model through an embedding layer. Considering the influence of the time interval between consecutive time steps on distinguishing the normality of behaviors, we further modify the traditional LSTM by feeding the time interval between two consecutive time steps into the forget gate to capture the nonfixed time interval effect. Furthermore, to allow for the long-term dependency of the behaviors and to capture periodic or repeated behavior, we further employ a self-historical attention mechanism for the sequence of behavior to capture the possible cyclical and repeated behavior in the normal behavior sequences, in which each historical hidden layer is employed to attend the representation of the current time step. Moreover, in regard to the interaction structure manifested in the dynamic bipartite network, we further propose a network embedding approach on the basis of DeepWalk [1] by designing a novel random walk strategy according to the collaborative behaviors of a possible group of fraudsters. Then, the derived embedding of each user in the network can be trained together with the attributed behavioral sequence. Finally, we can derive the sequential behavioral embeddings of each user.

By utilizing the embeddings of the attributed behavior sequences, we can further identify fraudsters that have abnormal behavioral sequences. To validate the effectiveness of the learned behavioral embeddings, we implement the task of sequence prediction under the assumption that the future behaviors of normal users can be better predicted, while in contrast, fraudsters cannot be well predicted with greater values of loss. Furthermore, we can incorporate a set of ground truth labels in the model as positive instances to train the representations. We experiment on a real-world telecommunication dataset, and it shows that in both prediction-based and classification-based fraud detection tasks, the proposed model can achieve better performances to detect fraudsters. Additionally, through the visualization of attention weights for normal users and fraudsters, we can see that the proposed model can distinguish frauds from normal callers regarding their behavioral patterns with some reasonable interpretations.

2 Related work

In this paper, we aim to learn the embeddings from the sequential behaviors for fraud detection purposes. Therefore, our work is naturally related to fraud detection, RNN-based sequence models and attention models. Furthermore, since we incorporate the interaction network for modeling behavior, we also revisit the recent advancements in network embedding approaches.

2.1 Fraud detection

Fraud detection is one of the typical application scenarios of anomaly detection [2], which can be seen as distinguishing behavioral patterns of abnormal individuals that deviate significantly from those of normal individuals. Fraudulent instances are prevalently hidden inside financial transactions, telecommunication networks, health care insurances, online auctions, etc. [3]. In different application domains, the definition of fraud detection tasks can be varied in terms of how the fraudsters are defined. The most intuitive methods to tackle fraud detection problems are associated with assigning an anomaly score to each test instance. The anomaly scores are then used to rank the instances to discover the most anomalous ones. Ramaswamy et al. [4] proposed a KNN-based method to compute an anomaly score using the average distance of a point to its k -nearest neighbors. Yamanishi et al. [5] detected outliers for insurance based on a finite mixture model from the viewpoint of statistical learning theory. Hooi et al. [6] proposed a graph-based algorithm to spot fraudsters who published fake reviews in a social network. Robinson [7] proposed a method to detect fraud by using divergence analysis of the hidden Markov model when processing the stream of financial card transactions. Instead of using unsupervised methods and semisupervised methods, some methods also employ a classifier to assign a label to each test data instance. Bhattacharyya et al. [8] compared the performance of logistic regression, support vector machines, and random forests for detecting fraud on a real-world dataset of international credit card transactions, which yields precision of 0.366, 0.578, and 0.613. Tsang et al. [9] demonstrated that supervised learning methods can defeat some existing methods over real data and evaluated a set of features for a general fraud detection problem.

2.2 Recurrent neural networks

The recurrent neural network (RNN) is a feedforward neural network with the ability to capture the dynamics of sequential data by passing memory information across time steps [10] and has been widely applied in NLP [11], video caption [12], and recommender systems [13]. The recurrent structure of RNN endows it with good ability to store and upgrade sequential patterns, which could automatically learn the sequence representation. Chov et al. [14] proposed an RNN-based encoder-decoder framework to transform a varied-length sequence of symbols into a fixed-length vector representation, and the qualitative results of experiments show that the model learned a linguistically meaningful representation of phrases.

RNN has also been tailored to improve its effectiveness and efficiency to accommodate different scenarios. Collins et al. [15] designed a model called “the Update Gate RNN (UGRNN)” to build up the efficiency of training the RNN model. The UGRNN holds a concise architecture with only a coupled gate between the recurrent hidden states, which can improve the ease of training deep feedforward networks. Neil et al. [16] proposed a “Phased LSTM” model, which enhanced the long short-term memory (LSTM) cell by adding a new time gate for modeling event-based sequential data. Liu et al. [17] extended RNN as spatial temporal recurrent neural networks (ST-RNN) for modeling spatial and temporal contextual information that can help predict a user’s next action.

2.3 Attention mechanisms

The attention mechanism was first introduced to solve the alignment problem in the sequence-to-sequence models [18,19]. It provides a more flexible and accurate alignment instead of compressing the whole input sequence into a fixed semantic vector, and it is able to present the interpretability of dependencies between the input and output sequence. Not limited to the machine translation task, attention-based methods have been widely applied in many other domains. Due to the nature of the attention mechanism that aims to capture the close interrelationship between the latent semantic representation of various objects, it is very suitable for development in multimedia and heterogeneous data mining tasks [13,20]. Recently, some research works have utilized the attention-based sequential models to improve behavior prediction. Feng et al. [21] proposed a historical attention module to augment the recurrent neural network for multilevel periodicity prediction. To better understand the cascade dynamics, Wang et al. [22] adopted an attention-based RNN to solve the cross-dependence problem in cascade prediction.

2.4 Network embedding

Network embedding aims to find a low-dimensional vector space that can preserve properties of the original network [23]. Traditional network embedding works have been developed with general dimension reduction techniques [24–27]. However, these techniques usually suffer from high computational cost, making them impractical for dealing with large-scale networks. Recently, more efficient and effective network embedding methods have been proposed, which are inspired by recent advances in learning word embeddings from a collection of sentences [28]. For example, Perozzi et al. [1] utilized random walks to generate sequences of nodes in a large-scale network, and then considered the walking path as a sentence of words. Grover and Leskovec [29] extended the random walk procedure to a biased version, which can strike a balance between local and global properties of a network. Tang et al. preserved network structures by approximating first-order and second-order proximities in the embedding space [30]. Moreover, high-order proximities of nodes as well as community structures have also been taken into consideration in network embedding models [31].

3 Preliminary

In this section, we first conduct an exploratory analysis in a real-world telecom dataset to discover the behavioral patterns of fraudsters compared to those of normal users, and further exhibit the motivation of our model. Moreover, we will give an explicit definition of the problem.

3.1 Real-world dataset description

We obtained a real-world telecommunication dataset from one of the largest telecommunication operators in China. The dataset contains millions of call detail records (CDRs for short), with each recording the detail information of a phone call. The CDR includes information such as the phone numbers of both the caller and callee, the starting time of the call, the call duration and the call ending time. For privacy concerns, all the identifiable information of individuals (e.g., user's phone number, regions) is encrypted. With the anti-fraud service

provided by the telecom operator, receivers can label each incoming call as either frauds or not. For the concern of subjective judgment, we require that a caller needs to be labeled as fraud for more than ten times to be recognized as a fraud, i.e., only if at least 10 different calls launched by the same caller are labeled as fraud, the caller can be regarded as fraudster. This enables us to gather 13,028 labeled callers with 1,619,562 call detail records in total from April 1 to April 30, 2017. Callers that are relatively inactive, i.e., those who have less than 10 calling records in this time period, are filtered out. The number of fraudsters is 428, accounting for approximately 3% of the dataset. Further, we gather all the callers' full calling sequences in this period, which gives us 1,619,562 call detail records with more than 300,000 distinct callees.

3.2 Analysis of sequential interactive behaviors

Given the real-world telecom dataset, we intend to reveal the behavioral differences between fraudsters and normal users from the following perspectives: (1) Do they have a specific time regularity in their routine activities? (2) Are there any unique sequential behavioral patterns for fraudsters? (3) How do the callers interact with their target callees, and how does the interaction network differ from that of the normal callers?

The temporal mode of fraudsters We first calculate the distribution of calling times for normal users and fraudsters. As Fig. 1a presents, almost all of fraudsters' calls occur between

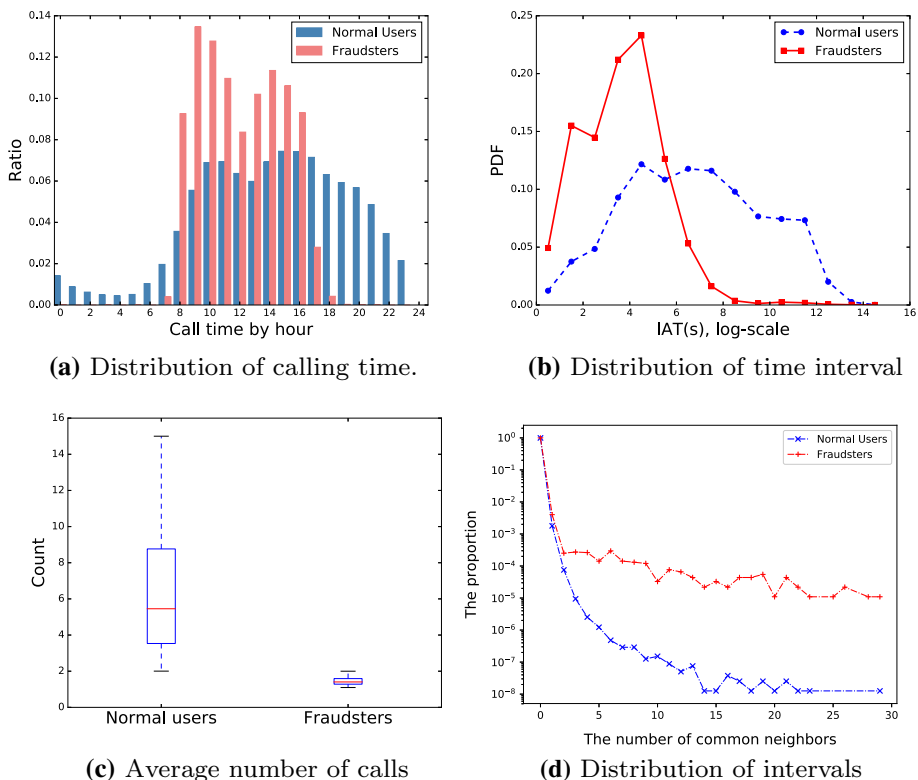


Fig. 1 Exploratory analysis of behaviors in telecom networks

8:00 a.m. and 6:00 p.m., and the maximum number of calls are made at 9:00 a.m. However, many normal users still make many calls after 6:00 p.m. and have a small number of calls before 8:00 a.m., which shows that fraudsters' calling behaviors may follow the mode of the working clock. This observation further implies that normal users consider the telephone a communication tool in their daily routine; however, fraudsters treat the telephone as the main instrument of their work and manifest the specific temporal mode of their routine calling activities. To this end, it is worth considering incorporating the property of calling time to uncover the fraudsters' unique behavioral patterns.

The sequential pattern of calling behaviors To further uncover the divergence of sequential patterns between normal users and fraudsters, we show the probability density function (PDF) curve of the interarrival time (IAT) between their consecutive behaviors. As shown in Fig. 1b, the horizontal axis represents the logarithm values of consecutive IATs, and we can see that most fraudsters make two consecutive calls within 1 h and that most of their time intervals centralize around 1 min, which illustrates the phenomenon that fraudsters tend to make intensive calls. However, normal users' time intervals span a wider range from approximately a few seconds to almost 2 days. As a matter of fact, frequent and short time intervals of the sequence can reveal that the calling behaviors are coherent, planned and an intensive activity, with every two adjacent calling behaviors having a strong consistency with each other. In contrast, long time intervals mean that the necessary associations between the behaviors may not exist.

The dispersion of interactions In telecommunication networks, each node represents a user, and the edges represent the calling interactions between users. Each edge can have an edge weight representing the intensity of calling interactions between the source and target pair, and a larger weight can indicate a larger number of calls between the pair. We then calculate the average weight of each individual caller in the network (i.e., the sum of weights on all the edges of a node divided by the degree of the node), which can reflect the dispersion of target contacts in the interaction network. We further draw a box-plot distribution of the average weights for fraudsters and normal users. As shown in Fig. 1c, compared with the fraudsters' box-plot, the normal users' box-plot has a much higher median and a much wider distribution range, meaning that normal users maintain a close and stable connection with their neighbors. In contrast, the box-plot of fraudsters is compressed into a flat box at the bottom, which shows that fraudsters constantly dial new contacts and seldom re-call their previous contacts, thus indicating a high rate of dispersion in the interaction structure.

The interaction structure Given the bipartite interaction network, in order to show the differences of the fraudsters and normal users with regard to their interaction structure and the closeness between different source users, we calculate the number of common neighbors between each pair of fraudsters and normal users, respectively. The distribution of the number of common neighbors is shown in Fig. 1d. It is interesting to observe that fraudsters generally have more common neighbors than normal users; i.e., different fraudsters are more likely to call the same contacts more frequently than the normal users do. This may be possible because fraudsters may work in groups to target a focal group of target users, in which each fraudster may play a specific role and work collaboratively according to a predesigned script.

Based on the above analysis, we can make the following observations which provide modeling intuitions in representing the sequential behaviors for the purpose of distinguishing fraudsters from normal users.

- Fraudsters' calling activities follow the temporal regularity of the working mode.
- There are significant differences between normal users and fraudsters in terms of their sequential patterns, particularly the time interval between consecutive calling activities.

- Normal users maintain close interactions with a set of stable contacts, while fraudsters constantly call new contacts.
- The closeness between fraudsters differs tremendously compared with that between normal users. In other words, the higher-order similarity of nodes can be exploited to distinguish frauds.

3.3 Problem definition

In telecommunication networks, each call consists of two parties, i.e., a source user and a target user. For the purpose of fraud detection, we focus on the source, who initiates the calling behavior, and then target users, along with other critical characteristics to depict the calling behaviors, can be regarded as specific attributes at each time step in the behavioral sequences. Formally, given a temporal directed bipartite graph $\mathcal{G} = \langle \mathcal{U}, \mathcal{V}, \mathcal{E}; \mathcal{A}, \mathcal{P} \rangle$, $\mathcal{U} = \{u_1, \dots, u_n\}$ denotes the source nodes and $\mathcal{V} = \{v_1, \dots, v_m\}$ denotes the target nodes of the network. Each source user $u \in \mathcal{U}$ has a feature vector $\mathbf{p}_u \in \mathcal{P}$, and $e_{uv}^{(t)} \in \mathcal{E}$ denotes the edge between source node u and target node v formed at time t , which corresponds to the interaction behavior at time t , and the attribute vector $\mathbf{a}_{uv}^{(t)} \in \mathcal{A}$ denotes the characteristics of the interaction behavior of u and v at time t . Thus, for each source node u , the behavioral sequence denotes $q^u = [(t = 1, v^{(1)}, \mathbf{a}_{uv}^{(1)}), (t = 2, v^{(2)}, \mathbf{a}_{uv}^{(2)}), \dots, (t = T, v^{(T)}, \mathbf{a}_{uv}^{(T)})]$, where T is the maximum time steps of the sequences.

4 The proposed model

According to the above analysis, we observe that fraudsters can diverge from normal users in terms of both sequential regularity and the interaction mode. Therefore, motivated by these observations, we propose to encode sequential behaviors by combining both the attributed behavioral sequences with the dynamic interaction structure. Then, we can further employ the learned behavioral representations for sequence prediction and classification to detect fraudulent behaviors.

Specifically, each individual has an attributed behavioral sequence; therefore, it is naturally appealing to feed the sequence into an LSTM [32], a state-of-the-art sequence learning method to learn the representations. Considering the influence of different time intervals on distinguishing behavioral patterns, we can augment the basic recurrent unit with time intervals. Additionally, a *self-historical attention module* is developed to allow repeated and periodic occurrences of interaction events to capture particular routine temporal patterns, and those that diverge from the routine patterns have the potential to be identified as frauds. Moreover, the interaction structure can be further exploited for modeling the behavioral sequences. In our prior work [33], in order to consider the interactions between sources and targets, we feed the source and target users to an embedding layer and construct the *interaction module*, which yields the model **HAInt-LSTM**. In this paper, considering the differences in closeness in the interaction networks between fraudsters and normal users, we proceed to derive node embeddings from higher-order relationships in the interaction networks with a novel method **FraudWalk** and then replace the original ID-based embeddings to improve the *interaction module* for better representations of the sequential behaviors, which finally yields the **Network-enhanced Historical Attention-based-LSTM (NHA-LSTM)**. The model architecture is shown in Fig. 2.

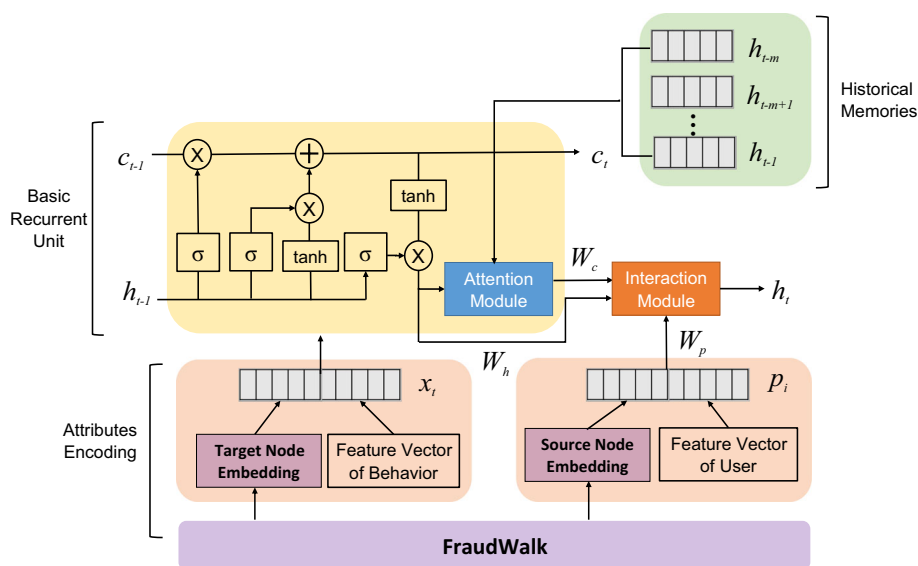


Fig. 2 Illustration of the proposed model NHA-LSTM

4.1 Recurrent unit with time interval

Given the behavioral sequences, we can apply the LSTM to learn the sequence embeddings. However, one major problem of an LSTM is that it records only the orders of the behaviors without considering the specific time interval, which limits the traditional LSTM in capturing the influences of different time intervals on representing sequential patterns. As witnessed in Sect. 3.2, fraudsters generally have a higher calling frequency in a short period of time that deviates from normal calling behaviors, and therefore, it is deemed desirable to take the time intervals between two consecutive behaviors to reveal the intensity and consistency of human activities.

Thus, we modify the original recurrent unit in the LSTM to capture the effects with respect to the varied lengths of the time intervals and store this time information until it is passed to the next recurrent unit. In particular, we add the term of time intervals $\Delta_{t-1,t}$ in the forget gate of the recurrent unit, with a set of parameters W_{ft} . Then, at each time step, except for receiving the previous hidden state and the current input vectors, we also consider the time interval from the last time step. In this regard, the time interval can play a role in determining whether the previous state should be stored or not. With the set of parameters concerning the forget gate learned from the time interval sequences, the states with time intervals that can reveal fraudulent behaviors would be more likely to be stored. Thus, specific sequential patterns with auxiliary time intervals can be captured.

By introducing the recurrent unit with time intervals, we can design the basic structure of the refined recurrent unit as follows:

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + W_{ft}\Delta T_{t-1,t} + b_f), \quad (1)$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i), \quad (2)$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o), \quad (3)$$

$$\tilde{c}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c), \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (5)$$

where x_t denotes the t -th sequence item vector. h_{t-1} denotes the $(t - 1)$ -th hidden state vector. $\Delta T_{t-1,t}$ denotes the time intervals between the timestamp of the t -th and the $(t - 1)$ -th behavior. f_t , i_t , o_t represent the forget gate, the input gate and the output gate of the t -th recurrent unit, respectively. W_{fh} , W_{ih} , W_{oh} , W_{fx} , W_{ix} , W_{ox} , W_{ch} , W_{cx} and W_{ft} are all the weight parameters, which project different input vectors and hidden states into the same latent geometric space. b_f , b_i , b_o and b_c are the corresponding biases. \tilde{c}_t is a new generated cell state, and c_t is the current updated cell state vector. \odot denotes the elementwise product.

4.2 Self-historical attention module

According to the analysis in Sect. 3.2, normal users tend to maintain stable and close interactions with others, which means they would repeat calling their previous contacts with a temporal mode or periodicity. However, fraudsters can hardly maintain long and stable relationships with targets since they constantly switch their calling behaviors to harvest more victims. Therefore, it is necessary to build the measurement of regularity and the relationship between the current behavior and the preceding behaviors. By employing the potential characteristics from previous behaviors as historical context information, we can capture the similarity of behaviors that occurred previously with the current behavior to enhance the representations, in which attention mechanism is appropriate for the task. Attention mechanisms have become a focal point of research interests in both academia and industry and were initially designed to solve the multidependencies and alignment problem of the source-to-target or target-to-source in language modeling [18]. The attention mechanism generally designs scoring functions [34] to measure the similarity between two objects.

Inspired by the philosophy of the attention mechanism, we design a *self-historical attention module* to capture the similarity of historical behaviors with the current behaviors to better capture repeated occurrences and possible periodicity. More formally, for the behavior at each time step t , we can track its preceding behavior at time step k within maximum M -length memories, i.e., $k \in \{t - M, t - M + 1, \dots, t - 1\}$, and the scoring functions and attention weights can be calculated through Eqs. (6)–(9).

$$\tilde{h}_t = o_t \odot \tanh(c_t), \quad (6)$$

$$e_{t,k} = W_\alpha \tanh(W_{\alpha s}s_t + W_{\alpha k}h_k + b_\alpha), \quad (7)$$

$$\alpha_{t,k} = \text{softmax}(e_{t,k}), \quad (8)$$

$$g(h) = \sum_{k=t-M}^{t-1} \alpha_{t,k} \odot h_k, \quad (9)$$

where \tilde{h}_t represents the candidate hidden state transformed by the current cell state. s_t is the t -th unit state of the recurrent neural network. In this model, we concatenate the candidate hidden state and cell state as the complete unit state. To implement the attention mechanism, we score each historical state h_k by comparing it with the current unit state s_t in Eq. (7) with a $\tanh(\cdot)$ function, where $W_{\alpha s}$ and $W_{\alpha k}$ are weight parameters for the current states and previous states, respectively, and W_α is the weight parameter for the time step and b_α is the attention bias to obtain the raw attention weight $e_{t,k}$. Further, we normalize the scores as the final output of attention weights with a softmax function and let $\alpha_{t,k}$ denote the normalized

weight. $g(h)$ represents the historical context vector which is the weighted average of all the historical states.

Along with the forward process of the LSTM, we can derive the hidden vectors h_k output from each previous time step, and take it to attend the current state representation s_t through a nonlinear scoring function in Eq. (7); the attention weights can then be derived by a softmax function (8).

4.3 Interaction module

Conventional sequence models focus on modeling the internal dependencies of different time steps but usually neglect the dynamic contextual information. However, as we have observed previously, not only the sequential behaviors but also the interaction structure can help distinguish fraudulence. As shown in Fig. 1c, fraudsters have tremendously more dispersed calling targets than normal users. Thus, we need to address how and toward whom the source users launch their activities, except for the time order of the activity. In particular, in our prior work [33], we consider the interactions between source and target users with an *interaction module*, in which we feed the IDs of source and target users through an embedding layer directly and concatenate them with their corresponding feature vectors, and then we can derive the modified behavior embeddings by combining the historical contextual information output from Eq. (9) as follows,

$$h_t = \tanh(W_h \tilde{h}_t + W_p \mathbf{p}_i + W_c g(h)), \quad (10)$$

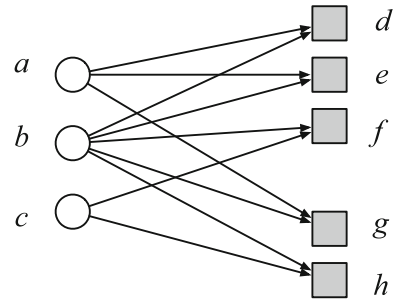
where \mathbf{p}_i is the personal feature vector of the i -th source user, and W_h , W_p , W_c are all weight parameters. We employ a multilayer perceptron (MLP) to obtain the t -th behavioral embedding h_t , with $\tanh(\cdot)$ as activation function. We first multiply the feature vector \mathbf{p}_i , the historical context vector $g(h)$ and the candidate hidden state \tilde{h}_t with the corresponding weight matrix, and take the sum as input for the MLP to generate the current behavioral embedding, which will be passed to the next recurrent unit.

4.4 Modeling higher-order interactions with network embedding

In addition to accounting for the direct connections between source and target users, fraudsters' behavioral patterns can also manifest in their higher-order relationships of the interaction network. The higher-order interactions indicate that source nodes share same similar first-order target nodes in the bipartite graph, which is similar to the concept of *guild* discussed in prior work [35]. As observed in Fig. 1d, fraudsters have higher ratios of common neighbors than normal users, showing that a fraudster may work closely with others to attack a focal group of targets, which we call *group fraudulence*. As a matter of fact, the interactions between source \mathcal{U} and target users \mathcal{V} can form a bipartite network \mathcal{G} . Such differences can be captured by considering higher-order relationships between the nodes in the network. Here, we use an example bipartite network to show how higher-order relationships can play a role in identifying potential group fraudulence, which is shown in Fig. 3, where nodes \mathbf{a} , \mathbf{b} and \mathbf{c} represent source users, while nodes \mathbf{d} , \mathbf{e} , \mathbf{f} , \mathbf{g} , and \mathbf{h} are target users.

As seen from Fig. 3, if we track along the calling records of users from the path “caller \rightarrow callee \leftarrow caller $\rightarrow \dots$,” the underlying relationships that cannot be directly observed from the first-order neighbors would be manifested. For example, by tracking the calling record of the caller \mathbf{a} , we can trace a target callee \mathbf{d} , and then by tracing back to another call received by \mathbf{d} , we can harvest another caller \mathbf{b} . Then, we can iteratively obtain \mathbf{f} and \mathbf{c} in

Fig. 3 Example of group fraudulence in a bipartite network



a row. Similarly, if we start from c , we can also obtain a path “ $c \rightarrow h \leftarrow b \rightarrow e \leftarrow a$.” This well demonstrates how the closeness between nodes can be revealed from higher-order relationships. As a matter of fact, we can see that a and c have no common neighbors if only the first-order relationships are considered, but in contrast, they are tightly coupled with b to collaboratively form a group in higher-order relationships.

Considering the real telecom scenarios, if the source users such as a , b , and c call the same group target user consecutively within a short time interval, they would have a higher potential to be identified as *group fraudulence*. Thus, it is desirable to consider the closeness between the source users and represent their higher-order interaction structure for probing fraud more accurately.

DeepWalk [1] is a commonly used method to derive node embeddings with regard to higher-order relationships in the network. However, DeepWalk allows the random walk to go through any two nodes that have links, which may be unsuitable for representing the potential group fraudulent behaviors. In fact, fraudsters who work as a group often launch their active behaviors in short time intervals toward a particular set of target users. Therefore, it is not appropriate to allow the random walk to pass any two connected nodes, but instead the walk should be constrained within a short time window. Inspired by DeepWalk, we propose a novel embedding method, *FraudWalk*, with special concern for the group fraudulence scenarios. In this method, the walk also starts from a randomly chosen node, and a *constrained random walk* is performed on the basis of random walks with a time constraint δ_t . Formally, the FraudWalk procedure is presented in Algorithm 1.

The major subprocedure in FraudWalk is *ConstrainedRandomWalk*(v_i, δ_t, l), which indeed rests on the basis of the traditional random walk. Specifically, given a starting source node u_i , a connecting edge would direct the walk to a random target node $v_k \in \mathcal{N}(u_i)$. The interaction event between u_i and v_k is recorded to happen at the time t_{u_i, v_k} . Then, the walk

Algorithm 1 FraudWalk

Input: bipartite network $\mathcal{G} = \langle \mathcal{U}, \mathcal{V}; \mathcal{E} \rangle$, time constraint δ_t , embedding size d , walk length l , window size w

Output: matrix of node embeddings $\Phi \in \mathbb{R}^{|\mathcal{G}| \times d}$

- 1: Initialize the embedding Φ by sampling from $\mathbb{R}^{|\mathcal{G}| \times d}$
- 2: **for** $j = 0$ to $MaxIter$ **do**
- 3: Shuffle the nodes in the network $\mathcal{M}_j = \text{shuffle}(\mathcal{U} \cup \mathcal{V})$
- 4: **for** each $v_i \in \mathcal{M}_j$ **do**
- 5: $\mathcal{W}_{v_i} = \text{ConstrainedRandomWalk}(v_i, \delta_t, l, w)$
- 6: SkipGram(Φ, \mathcal{W}_{v_i})
- 7: **end for**
- 8: **end for**

can restart from v_k , and proceed to search for a random neighbor; but during this process, the walk should be constrained to a range such that the interaction events with node v_k can only take place within the time period $[t_{uv} - \delta_t, t_{uv} + \delta_t]$; otherwise, the walk would end. In this way, only the interaction events that occurred recently can be tracked as a walk for constructing the node embeddings, and the source users that work together frequently toward same target users might have a close embedding and can therefore help capture higher-order interactions to identify group fraudulence.

Given the generated random walks, we apply SkipGram to learn the representation of nodes, which is a language model that maximizes the co-occurrence probability among words (nodes) that reside in a window size of w , in a sentence (walk). Formally, the co-occurrence probability with an independent assumption can be written as the follows,

$$P(\{v_{i-w}, \dots, v_{i+w}\} \setminus v_i | \Phi(v_i)) = \sum_{j=i-w \& j \neq i}^{i+w} P(v_j | \Phi(v_i)). \quad (11)$$

where $\Phi(v_i) \in \mathbb{R}^d$ is the representation of the node v_i . As seen from the above equation, the general idea of SkipGram is to employ the representation of v_i to predict the “neighbors” in the random walk. Therefore, the learned representation can encode the higher-order proximity of the node. By iterating over all the possible collocations in our constrained random walk, we can obtain the log likelihood, which can be optimized efficiently with hierarchical softmax or negative sampling.

With the derived node embeddings $\Phi(v_i)$ from the proposed FraudWalk, we can feed the node embedding $\Phi(v_i)$ with the model by concatenating it with the attribute vectors \mathbf{p}_i . Compared to the previously proposed HAIInt-LSTM, we replace the trained embedding with the ID-based embedding vector. Unlike HAIInt-LSTM that updates the embedding in each epoch, we keep the node embedding unchanged during the training process in order to capture the interaction patterns in the networks.

4.5 Encoding attributed behavioral sequences

As discussed previously, the behaviors at each time can be represented by multidimensional attributes, which need to be encoded to feed into the model for learning the representations. Basically, there exist two types of vectors in depicting the behaviors, i.e., the behavioral attributes in calling the target users, as well as the profiling attributes of the source users. To better identify users’ behavioral characteristics, we retain only the attributes with discriminative power as input for the model.

In addition, we treat the continuous and categorical attributes differently as input for the model. For continuous attributes, we normalize the value of each attribute to be in the range between 0 and 1; while for categorical attributes, we convert each category to a low-dimensional embedding vector with an embedding layer when the number of categories is large, and we also convert the attribute with a small number of categories via one-hot encoding, and then input into our model.

4.6 Fraud detection based on sequence embedding

To fully validate the effectiveness of the learned behavioral sequence embedding for fraud detection, we propose two tasks based on the embeddings with regard to the availability of labels for frauds. In particular, we propose two learning frameworks, i.e., predicting the next

calling target without any ground truth labels, and classifying the sequences based on the given labels for frauds.

Calling target prediction In this framework, we detect fraudsters based on unsupervised sequence prediction. At each time step, we predict the caller's next target given the current embeddings derived from the behavioral sequences. Since we assume that normal users have temporal and interaction regularities inside their behavioral sequences, their sequence would thus be more easily predicted, with the loss value being relatively small. However, if behavioral sequences are generated by fraudsters who do not have stable calling contacts and would always change their temporal modes of calling behaviors, the prediction of future contacts would be more difficult and result in greater loss values. Specifically, the loss function of this framework can be formulated as follows:

$$c'_{t+1} = \sigma(W_t h_t + b_t), \quad (12)$$

$$L_q = -\frac{1}{N} \frac{1}{T} \sum_{n=1}^N \sum_{t=1}^T [c_t \ln(c'_t) + (1 - c_t) \ln(1 - c'_t)], \quad (13)$$

where c_{t+1} is the calling target of the $(t + 1)$ -th time step, and c'_{t+1} is the predicted target for the $(t + 1)$ -th time step from the learned embeddings. We use cross-entropy to obtain the loss for the prediction. T is the maximum sequence length, and n is the total number of sequences.

Fraudster classification In this framework, we are given a completely labeled dataset, with a small proportion being fraudsters that can be regarded as positive instances, while the remaining instances are negative. Then, we can build a classifier based on the behavioral sequences to train the embeddings according to the labels. In this regard, we add a softmax layer on the hidden representation of the last time step, so the recurrent neural network is utilized as a classifier and provides a possible label for this sequence. The loss function of this framework can be formulated as follows:

$$y' = \sigma(W_T h_T + b_T), \quad (14)$$

$$L_u = -\frac{1}{N} \sum_{n=1}^N [y \ln(y') + (1 - y) \ln(1 - y')], \quad (15)$$

where y is the ground truth label of the user, y' is the predicted label of the user, T is the maximum sequence length, and n is the total number of sequences. Then, given a sequence to be classified, we can calculate the probability of the sequence being positive, i.e., $y = 1$, based on the sigmoid prediction function (14), and then determine the sequence to be classified as fraud when the corresponding probability is larger than that being negative.

Training algorithm Algorithm 2 illustrates the training process of the model. We employ the Adam [36] optimization algorithm to minimize the loss function. In both learning methods, we adopt the cross-entropy [37] loss as the loss function, which is detailed in Eqs. (13) and (15).

Algorithm 2 Training algorithm of NHA-LSTM

Input: Training sequence samples \mathcal{Q} ; The samples' personal feature vectors \mathcal{P} ; The interaction network \mathcal{G} ; The network parameters: Θ ; The maximum number of epochs *epoch*; The maximum number of mini-batches *batch*

Output: Trained model parameters Θ

```

1: Train the node embeddings  $\Phi(v) \leftarrow \text{FraudWalk}(\mathcal{G}, \delta_t, d, l, w)$ 
2: Concatenate feature vectors  $\mathcal{P}$  with the node embeddings  $\Phi$ .
3: Prepare the training dataset  $\mathcal{Q}$  and  $\mathcal{P}$ 
4: Initialize the parameters  $\theta$ 
5: for each  $j \in \{1, 2, \dots, \text{epoch}\}$  do
6:   Shuffle the training dataset
7:   for each  $u \in \{1, 2, \dots, \text{batch}\}$  do
8:     for each  $q^u \in \mathcal{Q}, p_u \in \mathcal{P}$  do
9:       Compute the  $h_t$  by Eq. 10
10:    end for
11:    Compute the loss function of training samples according to Eqs. (13) and (15).
12:    Update the parameters  $\theta$  by Adam algorithm.
13:  end for
14: end for

```

5 Experiments

5.1 Experimental setup

According to the goal of fraud detection from behavioral sequences, the proposed model NHA-LSTM is trained based on two tasks, i.e., predicting the next calling target, and classifying the behavioral sequences given the labels. Thus, we implement the experiments based on the two types of training.

- *Prediction-based training*: In this experiment, we split each user's sequential calling records into two parts. The first 80% of each sequence is used to train the behavioral sequence embedding, and the remaining data are used to evaluate the prediction errors. To fully uncover the behavioral patterns and interactive impact of the telecom network, we adopt the most discriminative attributes to describe each calling behavior. Specifically, for the attributes of each calling behavior, we use *callee ID*, *area code ID*, *duration time*, and *calling time by hour* as the raw features. With regard to the source users, we employ *call ID* and *number of total calling targets* as their raw features. After procedures of embedding and normalization, we concatenate each attribute representation vector and input it to the model. As discussed in the above analysis, the higher the prediction errors that are output from the sequence prediction, the more likely the user is a fraudster.
- *Classification-based training*: In the classification-based experiment, we aim at evaluating the effectiveness of the sequence representation based on a labeled dataset. We add a softmax layer to the last hidden output vector of each sequence and directly predict the label of the entire sequence.

Parameter Settings In the experiments, we set the mini-batch size as 1000, and the learning rate of Adam as 0.003. For the vector sizes, we set the size of attention vectors as 64, and the size of hidden state vectors is 64. In addition, the embeddings size for the trained network embedding FraudWalk is set as 256, with the time constraint $\delta_t = 3$ days, and the length of historical memories is 50. We apply gradient clipping during the training process, and the value of the gradient clip is 5.0. All the manually set parameters are shared by the two learning settings, and the other parameters are defined by default.

Baseline Methods We compare our method with several sequence learning methods, including LSTM and its variates. In addition, we also compare the proposed NHA-LSTM with our previously proposed model HAIInt-LSTM to validate the effectiveness in considering the higher-order interaction structure. Corresponding to the proposed model, we also train the sequences in both unsupervised and supervised settings. The baseline methods are detailed as follows.

- Long Short-Term Memory (LSTM) [32]: This method was originally designed to solve the gradient vanishing and explosion problem of the vanilla RNN. Its recurrent unit consists of a memory cell and three gates, namely input gate, forget gate and output gate, which help it to better model long-term dependencies.
- Gated Recurrent Unit (GRU) [14]: This method is another widely used variate of RNN. Compared with LSTM, it replaces the forget gate and the input gate with a single update gate and passes the hidden state directly to the next unit, while LSTM uses the output gate to wrap the hidden state. Moreover, it has a reset gate to control the information from the last moment.
- Update Gate RNN (UGRNN) [15]: This method makes a compromise between the vanilla RNN and LSTM/GRU, which uses a gate to determine whether the hidden state should be computed immediately or passed on. It implements the recurrent neural network with a succinct architecture, and improves learning performance and trainability for long sequences.
- Phased LSTM: The phased LSTM model is proposed to capture event-based sequential data with different time intervals. It extends the LSTM unit by adding a new time gate, which controls when the model should update the cell state and hidden state, and it has the other parameters to manage the phase shift and the percentage of activation time to the full time period [16].
- HAIInt-LSTM [33]: We propose HAIInt-LSTM to represent the behavioral sequences for fraud detection. In designing the interaction module, we only take the original IDs of source and target users as input, and concatenate them with the feature vectors as the input for the model.

Evaluation Measures We attempt to evaluate the learned embeddings for fraud detection based on both prediction and classification tasks. In fraud detection applications, the dataset is usually imbalanced. Generally, the most frequently used criteria for such imbalanced datasets include the F-measure, receiver operating characteristics (ROC) curves, cost curves and so on [38]. In this paper, we apply the F-measure and the area under the ROC curve (AUC) [39] as the evaluation measures of the experimental results.

All the experiments are carried out on a Debian 8 server with 24 2.4 GHz CPUs and 2 Tesla K40m GPUs.

5.2 Experimental results

5.2.1 Fraud detection based on target prediction

After training the model based on the prediction of calling targets as stated in Algorithm 2, we apply the trained model on the testing set to obtain the averaged loss, i.e., the average errors in predicting the calling targets of each individual. The callers with large average prediction errors are deemed to deviate from regular sequential patterns and are therefore more likely to be identified as fraudulent users. Then, by sorting the prediction errors in a descending order, we can obtain the ROC curves and calculate the AUC values. In general, the greater the

Table 1 The performance of fraud detection based on target prediction

Metrics	NHA-LSTM	HAInt-LSTM	LSTM	GRU	Phased-LSTM	UGRNN
AUC	0.8476	<u>0.8429</u>	0.8264	0.8225	0.8293	0.8106

The value in bold denotes the best performance, and the value underlined denotes the second-best

AUC value is, the more likely that the model would rank the true fraudulent samples ahead of the negative samples. As given in Table 1, our model achieves the best performances on the testing dataset in comparison with other baseline methods. It is also worth noting the previously proposed method HAIInt-LSTM achieves the second best performance but cannot compete with the newly proposed network enhanced model, showing the effectiveness of the framework and the strength in employing the higher-order relationships in the interaction network.

Among the other baseline methods, the phased LSTM performs better than others, which ensures that the time periodicity information is indeed helpful to model the behavioral patterns. This result also suggests that it is necessary and effective to integrate the behavior dynamics and context modeling, which enhances the performance of our model. However, without the specific design for long-term dependencies, GRU performs worse than the traditional LSTM model. Among all the baseline methods, UGRNN cannot defeat the others, showing that the time-related or context-related information is critical for sequence embedding.

5.2.2 Fraud detection based on classification

Given the available labels for fraudsters and normal users, we can proceed to validate the effectiveness of sequence embeddings based on classification by predicting class labels from the softmax layer. In reality, the distribution of categories in fraud detection applications is usually skewed, with only a small proportion of instances being frauds. Thus, in order to validate the sensitivity and robustness of the proposed model, we vary the fraud proportion in the training dataset from 0.3 to 1% and test on a fixed dataset.

As Table 2 shows, from the perspective of the F1-score, the proposed model in this paper performs the best in the four different settings, while our previously proposed method HAIInt-LSTM achieves the second best, which demonstrates the advantages in introducing the higher-order relationships by the network embedding method. In particular, when fraud is extremely scarce in the training set, such as when the proportion is 0.3%, the F1-score of the positive class reported by our model still remains above 0.5 and the AUC value is above 0.95, which indicates that our model has good adaptability and stability in addressing extremely imbalanced datasets and shows potential to apply our model in real-world scenarios. It is also notable that when the proportion decreases to 0.5%, HAIInt-LSTM has the best AUC values, and while the performances of NHA-LSTM are slightly poorer, it still outperforms the other baseline methods.

Among all the baseline methods, the phased LSTM performs better than the others. As the results show, the AUC values of the phased LSTM always remain above 0.85; however, when the fraud proportion setting is 0.5%, the F1-score of the positive class by the LSTM approach is higher than the other baseline methods. However, none of the baseline methods can maintain a stable performance toward the ratios of frauds, which further reflects the robustness of our model framework by incorporating the temporal regularities and interaction module.

Table 2 The performance of fraud detection based on classification

Fraud proportion	Metrics	NHA-LSTM	HAInt-LSTM	LSTM	GRU	Phased LSTM	UGRNN
0.010	F1-score ^a	0.9171	0.8006	0.5102	0.5079	0.4647	0.7012
	Macro-F1	0.9532	0.8888	0.7346	0.7332	0.7107	0.8336
	Micro-F1	0.9810	0.9588	0.9244	0.9236	0.9198	0.9391
	AUC	0.9845	0.9797	0.7759	0.9439	0.9654	0.9589
0.007	F1-score ^a	0.8943	<u>0.7375</u>	0.3584	0.3712	0.2885	0.5273
	Macro-F1	0.9399	<u>0.8539</u>	0.6536	0.6600	0.6169	0.7414
	Micro-F1	0.9745	<u>0.9466</u>	0.9051	0.9055	0.8985	0.9187
	AUC	0.9649	<u>0.9390</u>	0.8004	0.9165	0.9313	0.8043
0.005	F1-score ^a	0.8602	<u>0.7160</u>	0.6417	0.2391	0.2099	0.2666
	Macro-F1	0.9210	<u>0.8427</u>	0.8012	0.5932	0.5781	0.6074
	Micro-F1	0.9680	<u>0.9447</u>	0.9293	0.9015	0.8994	0.9033
	AUC	<u>0.9556</u>	0.9646	0.9134	0.7735	0.9086	0.7472
0.003	F1-score ^a	0.7823	<u>0.5339</u>	0.1595	0.2195	0.1366	0.1762
	Macro-F1	0.8773	<u>0.7471</u>	0.5539	0.5849	0.5421	0.5626
	Micro-F1	0.9510	<u>0.9269</u>	0.9028	0.9067	0.9014	0.9039
	AUC	<u>0.9578</u>	0.9746	0.7201	0.7660	0.8576	0.7304

^a The F1-score of the positive class, i.e., the fraudulent class

The values in bold denote the best performance, and the values underlined denote the second-best

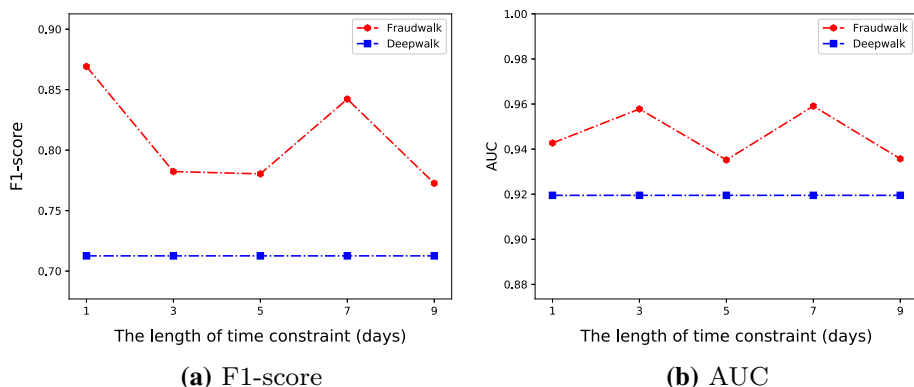


Fig. 4 Performances with varying time constraints for FraudWalk

To show the robustness of the proposed method, we also vary the parameter settings for the model. In particular, in FraudWalk, the time constraint δ_t plays a major role in identifying the structure of potential group fraudulence. Thus, we vary the value of the time to show how the time constraints improve the fraud detection performance. When the time constraints go to infinity, the embedding methods reduce to the traditional DeepWalk; therefore, we also treat DeepWalk as a baseline method and present the performances without the time constraint when training the node embeddings.

As shown in Fig. 4, the F1-score and AUC show slightly different trends. We can see that FraudWalk generally performs better than DeepWalk, which demonstrates the effectiveness in introducing the time constraint for learning the network representations. For the F1-score measurement, it performs best when the time constraint is set to be 1 day, and the longer time constraint may decrease the F1-score, but it still shows comparable performances when the time constraint is set to be 7 days. With regard to the AUC values, we can see that the time constraints of both 3 days and 7 days can achieve satisfactory performances, indicating that group fraudulence may be active in various time windows.

Moreover, we also vary the size of node embeddings to validate the performances of FraudWalk. As shown in Fig. 5, both the F1-score and AUC achieve the best performance when the embedding size is moderate with the value of 64.

5.3 Interpretability of attention weights

As described previously, normal users differ tremendously from fraudsters in terms of the behavior patterns, which motivates us to design the attention module in the model. To evaluate the effectiveness of the attention module and provide interpretability of the periodicity of user behaviors, we randomly sample 30 users from each class, and draw the normalized attention weights in a heatmap. Each row in Fig. 6 represents a single user, and the vertical axis denotes the attention weights at each time step. Darker colors in the grid denote larger attention weights. Generally, in the same scale of the color range, we find that most grids of normal users are light blue, which alternately emerge by a few deep blue grids, while the grids in the bottom heatmap are irregular and confusing. For example, the 14-th row in the heatmap of the normal users shows that every few grids there is a darker shade, which means it manifests very clear regularity and similarity within its behavior sequences. For different

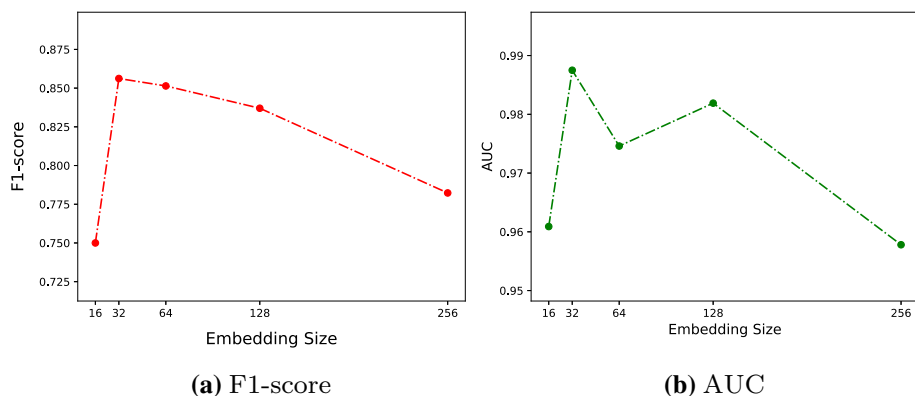


Fig. 5 Performances with varying node embedding sizes of FraudWalk

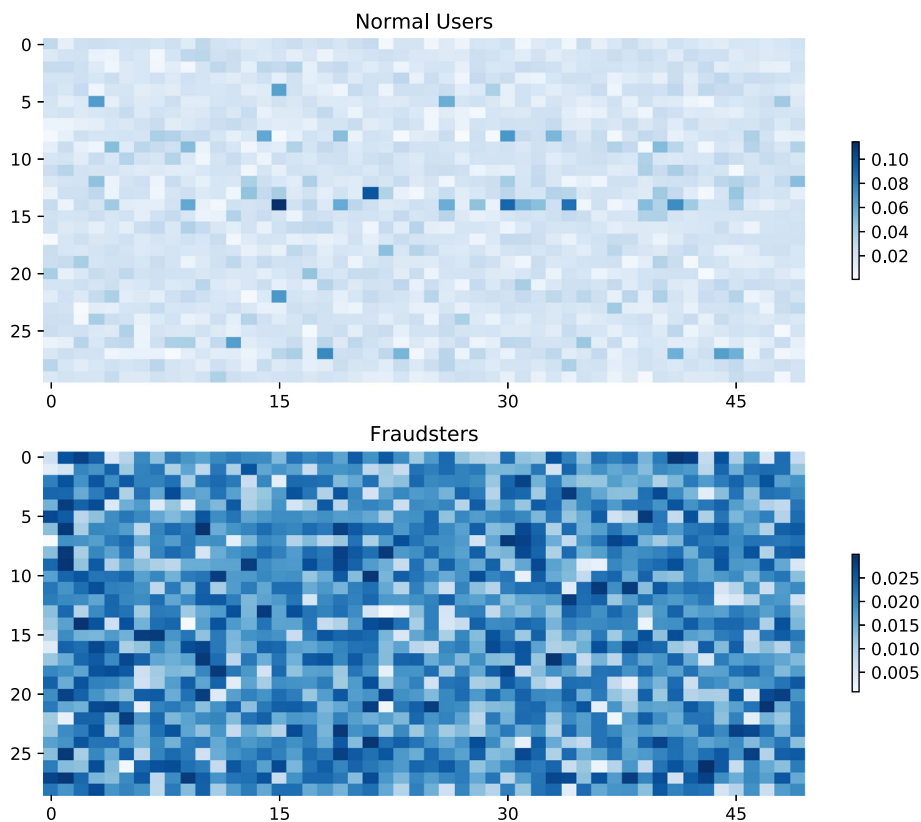


Fig. 6 Visualization of the attention weights (upper: normal users; lower: fraudsters)

users, the specific cyclical characteristics of behaviors are also varied. However, the heatmap of the fraudsters does not exhibit any prominent interpretable regularity, which illustrates that the behavior of the fraudster is a kind of randomness, and we find that their current

action poses difficulties for building similarity or periodicity characteristics by employing the historical behaviors.

6 Conclusion

In this paper, we propose a novel recurrent neural network called the Network-enhanced History Attention-based-LSTM (NHA-LSTM) model to learn sequential behavior representations for fraud detection. By incorporating the self-historical attention module and an enhanced interactive module with a novel embedding approach FraudWalk, NHA-LSTM achieves the best performances on target prediction and fraudster classification for fraud detection based on the learned sequential embeddings. In addition, according to the self-historical attention module, NHA-LSTM gains interpretability of the periodicities of human behaviors. Extensive experiments on a real-world telecom dataset demonstrate the superiority of our method over other sequence embedding methods for fraud detection.

In the future work, we can extend the modified forget gate to account for more complex time interval trends, since an increasing or decreasing trend of time interval can indicate possible fraudulent calling behaviors. For example, when a person is trapped to the fake story in the initial stage, more intensive calls with decreasing time intervals may be made to induce the person.

References

1. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: SIGKDD. ACM, New York, NY, USA, pp 701–710
2. Phua C, Lee V, Smith K, Gayler R (2010) A comprehensive survey of data mining-based fraud detection research. CoRR, vol. abs/1009.6119
3. Abdallah A, Maarof MA, Zainal A (2016) Fraud detection system: a survey. J Netw Comput Appl 68:90–113
4. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data, pp 427–438
5. Yamanishi K, Takeuchi JI, Williams G, Milne P (2004) On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. Data Min Knowl Discov 8(3):275–300
6. Hooi B, Shin K, Song HA, Beutel A, Shah N, Faloutsos C (2017) Graph-based fraud detection in the face of camouflage. ACM Trans Knowl Discov Data 11(4):1–26
7. Robinson WN, Aria A (2018) Sequential fraud detection for prepaid cards using hidden markov model divergence. Expert Syst Appl 91:235–251
8. Bhattacharyya S, Jha S, Tharakunnel K, Westland JC (2011) Data mining for credit card fraud: a comparative study. Decis Support Syst 50(3):602–613
9. Tsang S, Koh YS, Dobbie G, Alam S (2014) Detecting online auction shilling frauds using supervised learning. Expert Syst Appl 41(6):3027–3040
10. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. CoRR, vol. abs/1506.00019
11. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. CoRR, vol. abs/1409.3215
12. Li X, Zhao B, Lu X (2017) Mam-rnn: multi-level attention model based rnn for video captioning. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, pp 2208–2214
13. Zhai S, Chang Kh, Zhang R, Zhang ZM (2016) Deepintent: learning attentions for online advertising with recurrent neural networks. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1295–1304
14. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, vol. abs/1406.1078

15. Collins J, Sohl-Dickstein J, Sussillo D (2016) Capacity and trainability in recurrent neural networks. *CoRR*, vol. abs/1611.09913
16. Neil D, Pfeiffer M, Liu SC (2016) Phased lstm: Accelerating recurrent network training for long or event-based sequences. *CoRR*, vol. abs/1610.09513
17. Liu Q, Wu S, Wang L, Tan T (2016) Predicting the next location: a recurrent model with spatial and temporal contexts. In: *Proceedings of the thirtieth AAAI conference on artificial intelligence*, pp 194–200
18. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *CoRR*, vol. abs/1409.0473
19. Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y (2015) Attention-based models for speech recognition. In: *Proceedings of the 28th international conference on neural information processing systems*, pp 577–585
20. Chen J, Zhang H, He X, Nie L, Liu W, Chua TS (2017) Attentive collaborative filtering: multimedia recommendation with item- and component-level attention. In: *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pp 335–344
21. Feng J, Li Y, Zhang C, Sun F, Meng F, Guo A, Jin D (2018) Deepmove: predicting human mobility with attentional recurrent networks. In: *Proceedings of the 2018 world wide web conference*, pp 1459–1468
22. Wang Y, Shen H, Liu S, Gao J, Cheng X (2017) Cascade dynamics modeling with attention-based recurrent neural network. In: *Proceedings of the twenty-sixth international joint conference on artificial intelligence*, pp 2985–2991
23. Cai H, Zheng VW, Chang KC (2017) A comprehensive survey of graph embedding: problems, techniques and applications. *CoRR*, vol. abs/1709.07604
24. Kruskal JB, Wish M (1978) *Multidimensional scaling*. CRC Press, Boca Raton
25. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
26. Tenenbaum JB, Silva Vd, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323
27. Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *NIPS*. MIT Press, Cambridge, MA, USA, pp 585–591
28. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. *CoRR*, vol. abs/1301.3781
29. Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. In: *SIGKDD*. ACM, New York, NY, USA, pp 855–864
30. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: large-scale information network embedding. In: *WWW*. Republic and canton of Geneva, Switzerland: international world wide web conferences steering committee, pp 1067–1077
31. Wang D, Cui P, Zhu W (2016) Structural deep network embedding. In: *SIGKDD*. ACM, New York, NY, USA, pp 1225–1234
32. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
33. Guo J, Liu G, Zuo Y, Wu J (Nov 2018) Learning sequential behavior representations for fraud detection. In: *2018 IEEE international conference on data mining (ICDM)*, pp 127–136
34. Ma F, Chitta R, Zhou J, You Q, Sun T, Gao J (2017) Dipole: diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1903–1911
35. Alterovitz G, Ramoni MF (2006) Discovering biological guilds through topological abstraction. In: *AMIA annual symposium proceedings*, vol 2006, p 1. American medical informatics association
36. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *CoRR*, vol. abs/1412.6980
37. de Boer P-T, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. *Ann Oper Res* 134(1):19–67
38. He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284
39. Wu G, Chang EY (2005) Kba: kernel boundary alignment considering imbalanced data distribution. *IEEE Trans Knowl Data Eng* 17(6):786–795



Guannan Liu is currently an Assistant Professor in the Department of Information Systems with Beihang University, Beijing, China. He received the Ph.D. degree from Tsinghua University, China. His research interests include data mining, business intelligence, and anomaly detection. His work has been published in the journal of IEEE TKDE, ACM TKDD, ACM TIST, Decision Support Systems etc., and also in the conference proceedings such as KDD, ICDM, SDM, etc.



Jia Guo is currently a Ph.D. student at Department of Information Systems & Analytics, National University of Singapore. She received her Master degree from Beihang University in 2019. Her current research interests lie in explainable Artificial Intelligence, reasoning on knowledge graphs, and neural-symbolic integration.



Yuan Zuo received his Ph.D. degree from Beihang University, Beijing, China, in 2017. He is currently a post-doctor in Information Systems Department of Beihang University. His research interests include topic modeling and social computing.




Junjie Wu received his Ph.D. degree in Management Science and Engineering from Tsinghua University. He is currently a full Professor in Information Systems Department of Beihang University, and the director of the Research Center for Data Intelligence (DIG). His general area of research is data mining and complex networks. He is the recipient of the NSFC Distinguished Young Scholars award and the MOE Changjiang Young Scholars award in China.



Ren-yong Guo is currently a Professor at School of Economics and Management, Beihang University. He received his Ph.D. degree from Beihang University, China. Dr. Guo specializes in modeling and analyzing of pedestrian flow and network traffic flow assignment. He has published about 70 research papers in such journals as Transportation Research (Parts B and C), Transportation Science. His doctoral dissertation won the Award Nomination of the Excellent Doctoral Dissertation of China in 2011. He obtained the Program for New Century Excellent Talents in University of China in 2012. He obtained the National Natural Science Foundation of China for Excellent Young Scholars in 2016.

Affiliations

Guannan Liu¹  · Jia Guo² · Yuan Zuo¹ · Junjie Wu^{1,3,4} · Ren-yong Guo¹

Guannan Liu
liugn@buaa.edu.cn

- ¹ School of Economics and Management, Beihang University, Beijing, China
- ² School of Computing, National University of Singapore, Singapore, Singapore
- ³ Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China
- ⁴ Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations, Beihang University, Beijing 100191, China