Selective Graph Attention Networks for Account Takeover Detection

Jialing Tao
Platform Risk Management
Alibaba Group
Hangzhou, China
jialing.tjl@alibaba-inc.com

Wang Hui
Platform Risk Management
Alibaba Group
Hangzhou, China
saroe.wh@alibaba-inc.com

Tao Xiong
Platform Risk Management
Alibaba Group
Hangzhou, China
weilue.xt@alibaba-inc.com

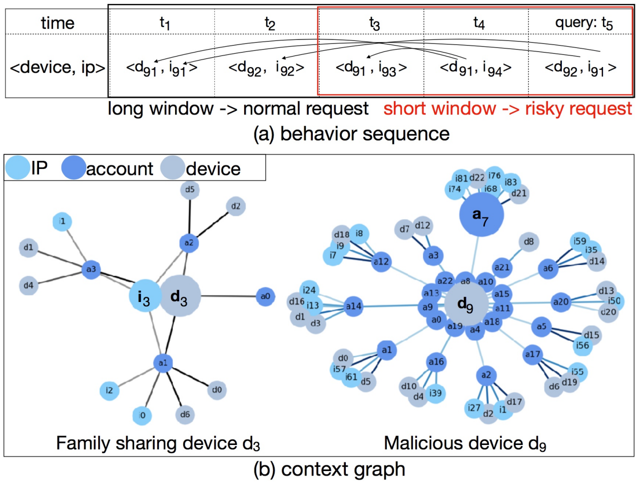
***Abstract*—Account takeover (ATO) is a type of fraud where a fraudster gains unauthorized access of a legitimate user's account through phishing, malware, bought credentials from dark web etc. The sophisticated evasion of detection by fraudsters and the requirement of friction-free experience by customers call for a new detection technique. Rather than using statistical features from behavior sequence in most of existing solutions, we represent account/context with graph node embedding and extract inherent sequential patterns with Recurrent Neural Networks (RNNs). Instead of using plain RNN by state-of-art fraud detection models, we further propose a selective graph attention mechanism within sequence (GAS), attending to only relevant steps to assist learning longer dependencies. The proposed selective graph attention mechanism is applicable to general graph structures. Experiments on a real dataset from Alibaba Group are conducted to compare the proposed model to several state-of-the-art approaches. The results demonstrate the effectiveness of the proposed graph attention mechanism. A real case study is also presented to further explain how proposed GAS works.**

***Keywords*-Account Takeover, Graph Attention, LSTM, Sequence Tagging, Fraud Detection**

I. Introduction

Account takeover (ATO) is a type of fraud where a fraudster gains unauthorized access to or even take full control of the accounts of legitimate users. It is an upstream unit of various fraud schemes, including account abuse, identity theft, etc. ATO is not only a threat to users but also brings financial and reputational loss to online service providers. The identity fraud study [1] released by Javelin Strategy Research found that $16 billion was stolen from 15.4 million U.S. consumers in 2016.

Nevertheless, there exists few research on ATO detecion even several models have been proposed for relevant tasks, e.g. access control, fraud detection. Most works either adopt detailed investigation of compromised accounts, or design handcraft statistical features. Typical features include behavior frequency, frequent action sequence of victims, node degree and edge weight in graph ([2]–[5]). Handcraft features are effective, but relying heavily on domain expertise which is limited to fraud schemes already known, and thus suffering from performance decay when facing adversary attempts. Recently, the state-of-art model [6] uses LSTM to model session-based click sequence to detect fraudulent purchase. Though LSTM alleviates the problem of learning long-term dependency, the model may still suffer from forgetfulness when dealing with long sequence, and we believe that attention augmented RNN can further improve the model on accuracy. To sum up, ATO detection remains challenging for the following two reasons.



(a) behavior sequence

(b) context graph

Figure 1. Examples illustrating the necessity of considering long sequence (a) and context graph (b)

Necessity of considering long sequence: The trend that users login across multiple contexts (e.g., devices, IPs) makes it harder to discriminate a malicious request from a normal request coming from a new context, and the subsequent increase of false positives causes burdensome authentication. Aside from the trend of legitimate users, more stolen accounts wait for weeks, pretending to be normal user to earn trust before attack, which makes it even harder for ATO detection. Fig. 1 (a) lists a behavior sequence of five steps, and each step is structured as a triplet of contexts (device d, IP prefix i). To decide whether to accept the query step t_5, the decisions made considering long and short windows are different. If considering short window (t_3 to t_5), the requester will be challenged with authentication due to inconsistency. While the request should be approved, as the contexts are actually consistent within a

2375-9259/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDMW.2018.00015

longer window (t_1 to t_5). We link the steps of same device or IP prefix in the sequence, and the linkages inspire us to use selective graph attention network within sequence to assist learning long term dependency in our work.

Necessity of considering context graph: Fraudsters no longer attack alone with a single stolen account, but create an army of accounts to make the fraudulent practices economically viable. Thus malicious devices are usually found connected to many accounts. However, family sharing devices and public devices can hardly be discriminated from malicious devices with only statistical features like node degree in hand. A fine-grained level of context graph is necessary for detection. Fig. 1 (b) shows the difference between normal device d_3 and malicious device d_9 in context graph, and each graph include nodes within two hops from the studied device. Edge color represents the intensity of connection, and node color and node name represent node type (a for account, i for IP prefix, and d for device ID). Both d_3 and i_3 are strongly connected to the same four accounts, and thus they are inferred as family sharing context, and a co-occurrence from i_3 and d_3 will be judged as low risk. While d_9 is suspicious, as it is surrounded by many weakly connected accounts and one of them a_7 is a compromised account. To capture finer graphical significance, node embedding is utilized in our work.

To address the aforementioned challenges, we present an innovative solution to detect ATO fraud. We first represent accounts and contexts with node embedding, and then use LSTM to model long behavior sequence. We propose a new graph attention mechanism within sequence based on LSTM, linking relevant steps to the present request to assist learning long-term dependency. In summary, our main contributions are as follows: noitemsep

- We introduce a framework composed of node embedding and RNN for ATO detection, automatically capturing structural features of context graph and inherent temporal relationship of behavior sequence.
- We propose a directional graph attention based mechanism conjunct with LSTM, learning dependency despite of any distance through attending relevant steps forward within the sequence.
- Extensive experiments using real-world dataset from Alibaba Group demonstrate that the proposed model outperforms state-of-art models.

The rest of this paper is organized as follows: Sect. II describes the proposed model. Sect. III demonstrates our experimental results and a case study. We briefly review the related work and discuss the connections of proposed approach to it in Sect. IV. Finally, we conclude the paper in Sect. V.

II. Model Formulation

Given an interlinked sequence of previous visits, our task, a binary classification problem, is to predict whether a request for access to an account should be approved. Fig.2 shows the high-level overview of our model, and the key part GAS is described concisely in Algorithm 1.

Algorithm 1 Graph attention augmented RNN

```
Input:
    The sequence of visits $S = [s_1, s_2, ..., s_T]$;
    The adjacency matrix $G$ ($G \in R^{T \times T}$), providing the linkages between related visits in sequence $S$;
Output:
    Risk probability of the $t$-th visit, $p_t$;
Apply RNN processing $S$ to get hidden states $H = [h1, h2, ..., h_T]$ ($h_t \in R^d$);
for $k = 0$ to # of attention heads $K$ do
    for $i \leq t$ do
        if $G_{it} \neq 0$ then
            Compute the concatenation-based attention weights $\alpha_{it}^{(k)}$ as Equation (1) with trainable parameter $W^{(k)}$, $v^{(k)}$.
        end if
    end for
    Compute the attention weighted sum of linearly transformed hidden states $W^{(k)} h_i$
end for
Average the attention weighted sums of $K$ heads, $h'_t$ as Equation (2);
Adopt a FC layer on $h'_t$.
return $p_t$;
```

A. Raw feature space

The input sequence of visits is denoted by $S = [s_1, s_2, ...s_T]$. In each visit, s_t is a tuple of various string variables (IP address, geo-location, operating systems, TCP/IP connection parameters, and browser configuration, event details, etc.). With the information above, we generate device fingerprint as device ID, compute statistical features of historical co-occurrence of the requesting contexts and the account, and design high-level features based on malicious device profiling and IP explanation. Aside from hand craft features, the discrete variables (accounts and contexts) are represented with unsupervised node embedding based on the bipartite graph mentioned in Sect. I. Event detail variables include event type (login, password change, etc.), channel (app, web, QR code), user name type (nick, email, cellphone), and event ID is represented by one-hot feature. Finally, all the variables are projected to vectors, and are concatenated in s_t.

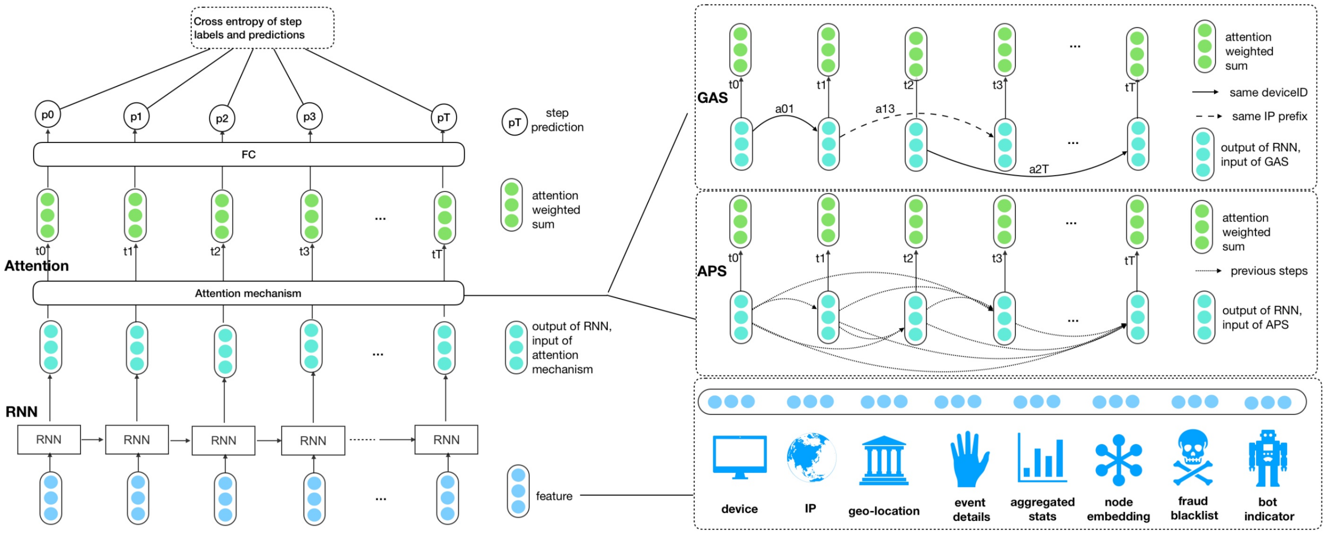


Figure 2. The Proposed Model Architecture. Features are concatenated embedding and stats as input, followed by a RNN layer. GAS is proposed to attend linked previous steps to the requesting step, unlike existing APS attends all previous steps. FC layer on top outputs the prediction

B. *GAS*

RNN (LSTM in our implementation) is used to capture dependencies in sequence.

$$h_1, h_2, ..., h_T = RNN(s_1, s_2, ..., s_T; \theta)$$

h_t denotes the hidden states of (the last layer) of RNN (if statcked) at the $t\,th$ step, and $h_t \in R^d$. θ is RNN's parameters. Since the sequence is usually long in our task, we use attention mechanism to assist learning long dependencies. Unlike existing state-of-art attention mechanism, which is to pay attention to all the previous steps, we attend to only targeted steps to further incorporate knowledge to the mechanism. The limitation of attention targets eliminates the potential distraction of attention to other irrelevant steps, and thus the accuracy of model can be improved. The links in sequence can be formulated as a directed graph, represented by an adjacency matrix G. In our task, an edge is established if the visits share the same device ID or IP prefix. Notice that the edges are directional, only from the present step to its previous steps, because only behavior in the past can be leveraged for decision. Also, the edges are unweighted, and thus the elements in G is binary. Considering the diversity of information encoded in sequence, we use multiple heads to allow attending from different perspectives, and the number of heads is denoted by K. For every attention head, we use a concatenation based attention mechanism to model the relation between hidden states of the two steps as (1) shows. For the $k\,th$ head, $W^{(k)}$ is a kernel applied on hidden states for linearly transformation, and $W^{(k)} \in R^{d' \times d}$. Non-linearity is applied on the concatenation of transformed states with activation function σ_1 (tanh in our implementation). $v^{(k)}$ is a trainable vector, and $v^{(k)} \in R^{2d'}$. We compute the attention weights of the $k\,th$ head $\alpha_{it}^{(k)}$, by normalizing the dot product of $v^{(k)}$ and the concatenation $[W^{(k)}h_i; W^{(k)}h_t]$ with the softmax function. In Equation (2), the attention weighted sum of each head is averaged to get a higher level representation of states h'_t, and $h'_t \in R^{d'}$.

$$\alpha_{it}^{(k)} = \frac{exp({v^{(k)}}^T \sigma_1([W^{(k)}h_i; W^{(k)}h_t]))}{\sum_{i \in N_i} exp({v^{(k)}}^T \sigma_1([W^{(k)}h_i; W^{(k)}h_t]))} \qquad (1)$$

$$h'_t = \sigma_2(\frac{1}{K}\sum_{k=1}^{K}\sum_{i \in N_i} \alpha_{it}^{(k)} W^{(k)} h_i) \qquad (2)$$

Different from existsing state-of-art attention mechanism in DIPOLE([7]) as following Equation 3, N_i in GAS (Equation (1)) denotes the set of first order neighbors of the $i\,th$ step (i.e., $g_{it} \geq 0$), while N in DIPOLE (Equation (3)) denotes the set of previous step indexes (i.e., $i \leq t$). Also the kernel matrix W is of a smaller size in our work ($W \in R^{2d \times d'}$ in DIPOLE), and the mechanism of multiple heads rather than single head is used in our work. When compared to the single head attention formulation in GAT ([8]), the object of non-linearity is different. GAT use LeakyRELU activation on ${v^{(k)}}^T[W^{(k)}h_i; W^{(k)}h_t]$, and the formulation of both parameters $v^{(k)}$ and $W^{(k)}$ transforming features of nodes seems redundant. Therefore, we apply activation on $[W^{(k)}h_i; W^{(k)}h_t]$ as in Equation (1), and $v^{(k)}$ is a necessary parameter vector for the generation of attention weights. Based on the same concatenation based attention mechanism (Equation (1) to (2)), we implemented GAS and a variant *Attention to all the Previous steps in Sequence* (APS) to verify the superiority of graph attention mechanism, and the results of comparison are interpreted with a detailed case study in Sect. III.

$$\alpha_{it} = \frac{exp(v^T \sigma(W[h_i; h_t]))}{\sum_{i \in N} exp(v^T \sigma(W[h_i; h_t]))} \quad (3)$$

Through a fully conneted (FC) layer, we obtain the risk probability p_t for the tth step.

C. Discussions

In combination with LSTM, GAS provides the teleportation capability to deal with long time sequence data. Imagine if we can teleport over an edge introduced by the shared IP address for example, the information propagation distance essentially shrinks. GAS gains great inspiration from previous graph embedding algorithms like GCN [9] and GAT [8]. Unlike GCNs and GATs which work with a global graph, GAS only applies graph attention within sequence, which we believe makes more sense for the ATO problem. If comparing GAS to APS, we suspect that the attention only over surely relevant steps rather than over all previous history can help ease the learning process and boost the prediction accuracy. The empirical results in Sect. III support this intuition.

III. Experiments

In this section, we evaluate the performance of the proposed model on a large real-world dataset, and illustrate the impact of GAS and APS on predictions with a case study.

A. Experimental Setup

1) Dataset: The dataset comes from one of the largest e-commerce platforms, Alibaba. The platform is subject to a barrage of fraudulent activities including but not limited to ATO, brushing orders, fake reviews, bot registration, malicious contents, coupon abuse, counterfeits, account selling etc. Since ATO fuels downstream fraud with compromised accounts, Alibaba devotes great resources for detection and has accumulated a high quality labeled dataset. The context graph for node embedding is established based on the successful access log on the platform of two months (Jul.-Sept. 2017). The dataset for node embedding has been pre-processed by filtering stop words(nodes) to preclude forgery/public device and NAT/proxy/cloud-host IPs. For the account reported and verified risky in Oct. 2017, the visits of the past two weeks before the risk report time have been labeled, and we use the dataset described above in our experiment. Table I lists the statistics of the dataset in our work.

2) Models for comparison: We compare five models as follows.

- Logistic Regression: Neglecting the temporal order of visits, every visit is treated individually for decision.
- LSTM: num of layer =1, d=64.
- LSTM-APS: num of layer =1, d=64, d'=8, K=2, attending to all previous steps.
- LSTM-GAS: num of layer =1, d=64, d'=8, K=2, attending to only linked steps.

Table I
Statistics of dataset

Statistics	Value
# of requests for train, val, test	848K, 94K, 448K
risk ratio of requests for train, val, test	10%, 10%, 0.5%
# of accounts for train, val, test	16019, 1784, 4571
Ave and max sequence length	50, 496
Ave link perc of risk and legit accounts	0.66, 0.75
vocab size of event ID	274
vocab size of account, device ID and IP prefix	186M, 268M, 2M

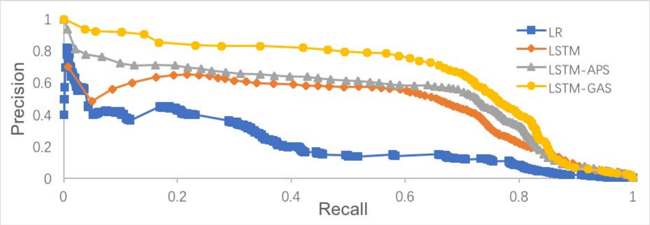


Figure 3. Precision Recall curves of Compared Models

3) Evaluation metrics: The performance of models are evaluated with F1 score, precision-recall curve is also provided for analysis.

4) Implementation details: We implemented all the approaches with Tensorflow. The hyper parameters are grid searched. The size of event ID embedding and node embedding are 8 and 22. The model was trained for 10 epochs using Adam optimizer with the batch size of 32 and the learning rate of 0.005.

B. Performance Evaluation

Table II shows the experiment results. LR underperforms the three temporal models, due to its constrained model complexity. Under the measure of F1 score, both attention augmented models predict more accurately than plain LSTM, benefiting from integrated information of previous steps for decision. With the same parameter size, LSTM-GAS is significantly superior than LSTM-APS in F1 score. In Fig. 3, the precision rate decreases dramatically at a early recall rate of 30% for LR, yet temporal models keep a steady precision till a recall rate of 60%, and the position of curves also demonstrated that LSTM-GAS achieves the best performance.

Table II
Evaluation of compared models

Model	F1 score	AUC	# of parameters
LR	0.3231	0.9201	326
LSTM	0.5762	0.9834	34853
LSTM-ATT	0.6127	0.9827	35796
LSTM-GAT	0.6834	0.9829	35796

For the analysis of the GAS mechanism, we define the link percentage in sequence as $\frac{l}{C_T^2}$, in which l denotes the

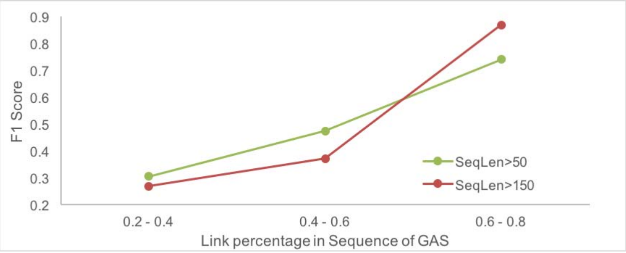


Figure 4. The Impact of Links in GAS on Accuracy

number of links (non-zero elements in G), and C_T^2 denotes the max number of links by connecting any two steps. As shown in Table I, the link percentage is usually higher for legitimate accounts than for victim accounts, since the consistency of context among steps lead to more connections in the sequence of legitimate accounts. Fig. 4 shows that the accuracy (F1 score) of LSTM-GAS model increases when link percentage gets large.

C. Impact of Attention Mechanisms

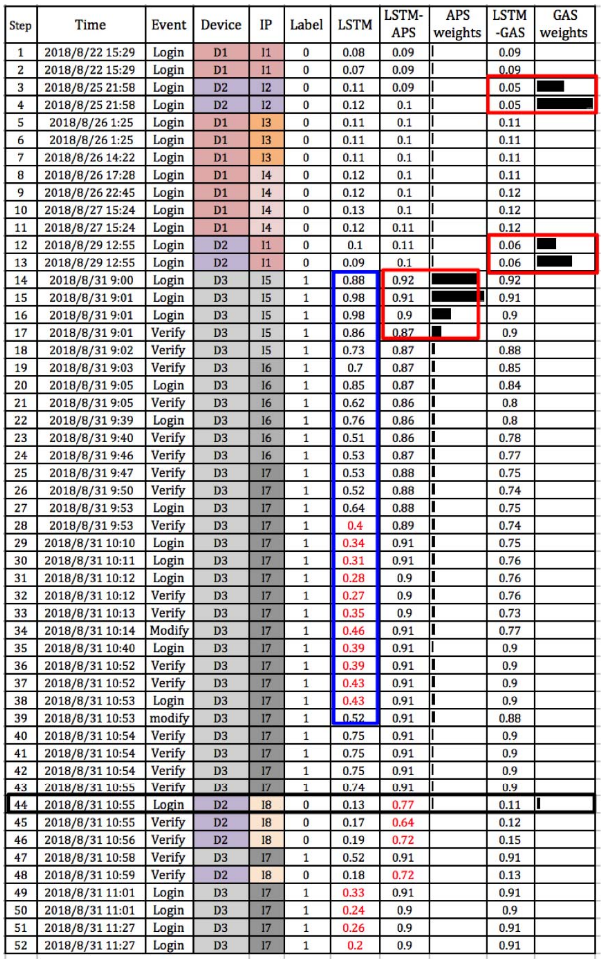
To further explain the impact of attention mechanisms, we predict a series of events of a stolen account with three models: LSTM, LSTM-APS, LSTM-GAS in Fig. 5.

LSTM performs the worst, suffering from a short memory of recent steps. From the observation of the blue frame, the LSTM model is able to detect the very first risky event, but it captures the consistence of recent steps as the perpetrator continues operating and thus outputs a wrong prediction of low risk afterwards. Compared with the decaying prediction score by LSTM in the blue frame, the outputs of both attention augmented models are steady in the same period. Moreover, LSTM-GAS outperforms LSTM-APS for the prediction of the $44\,th$ step. The predictions of most attended steps and the corresponding attention weights by LSTM-APS and LSTM-GAS are framed in red. It is a rare case that the control of account switches between of legitimate owner and the perpetrator, while for most of victim accounts the perpetrator seizes the control of account once successfully accessed to. It explains why LSTM-APS pays most attention to very first risky steps, and tends to score high once risk has been detected in previous steps. As for LSTM-GAS, with attention to only the steps sharing context, the predictions are more consistent for similar operations despite of temporal distance, and thus graph attention mechanism improves the model accuracy.

IV. Related Work

A. Attention Augmented RNN

Attention mechanism has become an integral part of state-of-art sequential modeling in conjunction with RNN, widely applied in tasks of text classification, question answering system, etc. Mimicking the attention system of human brain, the mechanism pays more or less attention to individual parts according to relevance. The attention mechanism further allows modeling dependency of any distance between, since direct attention shortens long-range sequential path length in the network, and thus attention augmented LSTM/GRU in general can achieve a better accuracy [10]. Also, its by-products, the attention weights, provide valuable insights about the specific parts contributing to the prediction decision, and thus improve model interpretability. Among many variants to generate attention weights, concatenation-based attention is commonly used and proved effective ([7], [8], [11], [12]) . Multi-head attention refers to a set of parallel individual attentions jointly attending information from different representation subspace, similar to human looking from multiple perspectives ([8], [10]).



Step	Time	Event	Device	IP	Label	LSTM	LSTM-APS	APS weights	LSTM-GAS	GAS weights
1	2018/8/22 15:29	Login	D1	I1	0	0.08	0.09	▮	0.09	
2	2018/8/22 15:29	Login	D1	I1	0	0.07	0.09	▮	0.09	
3	2018/8/25 21:58	Login	D2	I2	0	0.11	0.09	▮	0.05	███
4	2018/8/25 21:58	Login	D2	I2	0	0.12	0.1	▮	0.05	██████
5	2018/8/26 1:25	Login	D1	I3	0	0.11	0.1	▮	0.11	
6	2018/8/26 1:25	Login	D1	I3	0	0.11	0.1	▮	0.11	
7	2018/8/26 14:22	Login	D1	I3	0	0.11	0.1	▮	0.11	
8	2018/8/26 17:28	Login	D1	I4	0	0.12	0.1	▮	0.11	
9	2018/8/26 22:45	Login	D1	I4	0	0.12	0.1	▮	0.12	
10	2018/8/27 15:24	Login	D1	I4	0	0.13	0.1	▮	0.12	
11	2018/8/27 15:24	Login	D1	I4	0	0.12	0.11	▮	0.12	
12	2018/8/29 12:55	Login	D2	I1	0	0.1	0.11	▮	0.06	██
13	2018/8/29 12:55	Login	D2	I1	0	0.09	0.1	▮	0.06	████
14	2018/8/31 9:00	Login	D3	I5	1	0.88	0.92	█████	0.92	
15	2018/8/31 9:01	Login	D3	I5	1	0.98	0.91	██████	0.91	
16	2018/8/31 9:01	Login	D3	I5	1	0.98	0.9	██	0.9	
17	2018/8/31 9:01	Verify	D3	I5	1	0.86	0.87	█	0.9	
18	2018/8/31 9:02	Verify	D3	I5	1	0.73	0.87	▮	0.88	
19	2018/8/31 9:03	Verify	D3	I6	1	0.7	0.87	▮	0.85	
20	2018/8/31 9:05	Login	D3	I6	1	0.85	0.87	▮	0.84	
21	2018/8/31 9:05	Verify	D3	I6	1	0.62	0.86	▮	0.8	
22	2018/8/31 9:39	Login	D3	I6	1	0.76	0.86	▮	0.8	
23	2018/8/31 9:40	Verify	D3	I6	1	0.51	0.86	▮	0.78	
24	2018/8/31 9:46	Verify	D3	I6	1	0.53	0.87	▮	0.77	
25	2018/8/31 9:47	Verify	D3	I7	1	0.53	0.88	▮	0.75	
26	2018/8/31 9:50	Verify	D3	I7	1	0.52	0.88	▮	0.74	
27	2018/8/31 9:53	Login	D3	I7	1	0.64	0.88	▮	0.75	
28	2018/8/31 9:53	Verify	D3	I7	1	0.4	0.89	▮	0.74	
29	2018/8/31 10:10	Login	D3	I7	1	0.34	0.91	▮	0.75	
30	2018/8/31 10:11	Login	D3	I7	1	0.31	0.91	▮	0.76	
31	2018/8/31 10:12	Login	D3	I7	1	0.28	0.9	▮	0.76	
32	2018/8/31 10:12	Verify	D3	I7	1	0.27	0.9	▮	0.76	
33	2018/8/31 10:13	Verify	D3	I7	1	0.35	0.9	▮	0.73	
34	2018/8/31 10:14	Modify	D3	I7	1	0.46	0.91	▮	0.77	
35	2018/8/31 10:40	Login	D3	I7	1	0.39	0.91	▮	0.9	
36	2018/8/31 10:52	Verify	D3	I7	1	0.39	0.91	▮	0.9	
37	2018/8/31 10:52	Verify	D3	I7	1	0.43	0.91	▮	0.9	
38	2018/8/31 10:53	Login	D3	I7	1	0.43	0.91	▮	0.9	
39	2018/8/31 10:53	modify	D3	I7	1	0.52	0.91	▮	0.88	
40	2018/8/31 10:54	Verify	D3	I7	1	0.75	0.91	▮	0.9	
41	2018/8/31 10:54	Verify	D3	I7	1	0.75	0.91	▮	0.9	
42	2018/8/31 10:54	Verify	D3	I7	1	0.75	0.91	▮	0.9	
43	2018/8/31 10:55	Verify	D3	I7	1	0.74	0.91	▮	0.9	
44	2018/8/31 10:55	Login	D2	I8	0	0.13	0.77	▮	0.11	▮
45	2018/8/31 10:55	Verify	D2	I8	0	0.17	0.64		0.12	
46	2018/8/31 10:56	Verify	D2	I8	0	0.19	0.72		0.15	
47	2018/8/31 10:58	Verify	D3	I7	1	0.52	0.91		0.91	
48	2018/8/31 10:59	Verify	D2	I8	0	0.18	0.72		0.13	
49	2018/8/31 11:01	Login	D3	I7	1	0.33	0.91		0.91	
50	2018/8/31 11:01	Login	D3	I7	1	0.24	0.9		0.9	
51	2018/8/31 11:27	Login	D3	I7	1	0.26	0.9		0.91	
52	2018/8/31 11:27	Login	D3	I7	1	0.2	0.9		0.91	

Figure 5. Case Study of a Stolen Account: the predictions and attention weights by compared temporal models for a sequence of events.

The ATO detection task requires reviewing long-term behavior history. Therefore, attention augmented RNN is suitable to tackle the problem. We believe that the incorporation of knowledge to the attention mechanism by only attending to linked steps will result in a better accuracy of prediction on the present request, compared with the state-of-art mechanism attending to all the previous steps ([7]).

B. Graph Augmented Neural Networks

There have been several methods augmenting neural network with graph. Using a multi-step framework is a conceivable approach for augmentation, which is to train node embedding from the graph and then feed it to neural network as feature. There are also several attempts of graph convolution, encoding both local graph structure and features of nodes with sharing filters. Spectral and spatial methods are two common strategies to define convolutional filters. Spectral methods use a convolution operator defined in the Fourier domain with the eigen-decomposition of the graph Laplacian ([13], [14]). GCN [9] simplified the previous method via a first-order approximation of spectral graph convolutions. In spatial methods, kernels of fixed size are defined to provide filter localization for variable sized neighborhood. [8] proposed graph attention networks (GAT), extending masked self-attention layer from sequence-structured data to graph-structured data. Unlike GCN, GAT assigns different importances to nodes of a same neighborhood, enabling a leap in model capacity. And the size of kernel used in GAT to generate attention weights is fixed by nature, dropping limitation of neighborhood size. The multi-head attention mechanism allows multiple perspectives of localization, just like multiple filter mechanism in traditional Convolutional Neural Networks (CNN). Also, similar to average pooling in CNN, the attention weighted summations of hidden states of multiple heads are averaged to output a pooled representation.

The network of accounts and contexts is very helpful for ATO detection. Considering the large size of network and the subsequent huge number of parameters if trained jointly, we apply a two-step framework of unsupervised embedding first and training neural networks next.

V. Conclusions

In this paper, we present a graph attention based RNN approach for ATO detection. Empirical results on a real dataset from Alibaba Group demonstrate that our proposed model outperforms the popular state-of-the-art approaches. More generally, we believe the proposed GAS has potential to be applied to other sequence labeling problems, such as part of speech tagging, video classification etc.

References

[1] J. S. . Research, "2017 identity fraud study," News, Feb. 2017. [Online]. Available: https://www.javelinstrategy.com/press-release/identity-fraud-hits-record-high-154-million-us-victims-2016-16-percent-according-new

[2] A. Adler, M. J. Mayhew, J. Cleveland, M. Atighetchi, and R. Greenstadt, "Using machine learning for behavior-based access control: Scalable anomaly detection on tcp connections and http requests," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, Nov 2013, pp. 1880–1887.

[3] C. Liu and J. He, "Access control to web pages based on user browsing behavior," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, May 2017, pp. 1016–1020.

[4] H. Kim, S. Yang, and H. K. Kim, "Crime scene re-investigation: a postmortem analysis of game account stealers' behaviors," in *Network and Systems Support for Games (NetGames), 2017 15th Annual Workshop on*. IEEE, 2017, pp. 1–6.

[5] V. Frias-Martinez, J. Sherrick, S. J. Stolfo, and A. D. Keromytis, "A network access control mechanism based on behavior profiles," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*. IEEE, 2009, pp. 3–12.

[6] S. Wang, C. Liu, X. Gao, H. Qu, and W. Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 241–252.

[7] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "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*. ACM, 2017, pp. 1903–1911.

[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[11] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems*, 2016, pp. 3504–3512.

[12] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[13] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.