Modeling User Behavior With Interaction Networks for Spam Detection

Prabhat Agarwal
pagarwal@pinterest.com
Pinterest
USA

Manisha Srivastava
manishasrivastava@pinterest.com
Pinterest
USA

Vishwakarma Singh
vishwakarmasingh@pinterest.com
Pinterest
USA

Charles Rosenberg
crosenberg@pinterest.com
Pinterest
USA

ABSTRACT

Spam is a serious problem plaguing web-scale digital platforms which facilitate user content creation and distribution. It compromises platform's integrity, performance of services like recommendation and search, and overall business. Spammers engage in a variety of abusive and evasive behavior which are distinct from non-spammers. Users' complex behavior can be well represented by a heterogeneous graph rich with node and edge attributes. Learning to identify spammers in such a graph for a web-scale platform is challenging because of its structural complexity and size. In this paper, we propose *SEINE* (**S**pam D**E**tection using **I**nteraction **NE**tworks), a spam detection model over a novel graph framework. Our graph simultaneously captures rich users' details and behavior and enables learning on a billion-scale graph. Our model considers neighborhood along with edge types and attributes, allowing it to capture a wide range of spammers. *SEINE*, trained on a real dataset of tens of millions of nodes and billions of edges, achieves a high performance of **80**% recall with **1**% false positive rate. *SEINE* achieves comparable performance to the state-of-the-art techniques on a public dataset while being pragmatic to be used in a large-scale production system.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → *Web applications*; *Content ranking*.

KEYWORDS

Heterogeneous Graph Neural Networks, Spam, Machine Learning

ACM Reference Format:
Prabhat Agarwal, Manisha Srivastava, Vishwakarma Singh, and Charles Rosenberg. 2022. Modeling User Behavior With Interaction Networks for Spam Detection. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531875

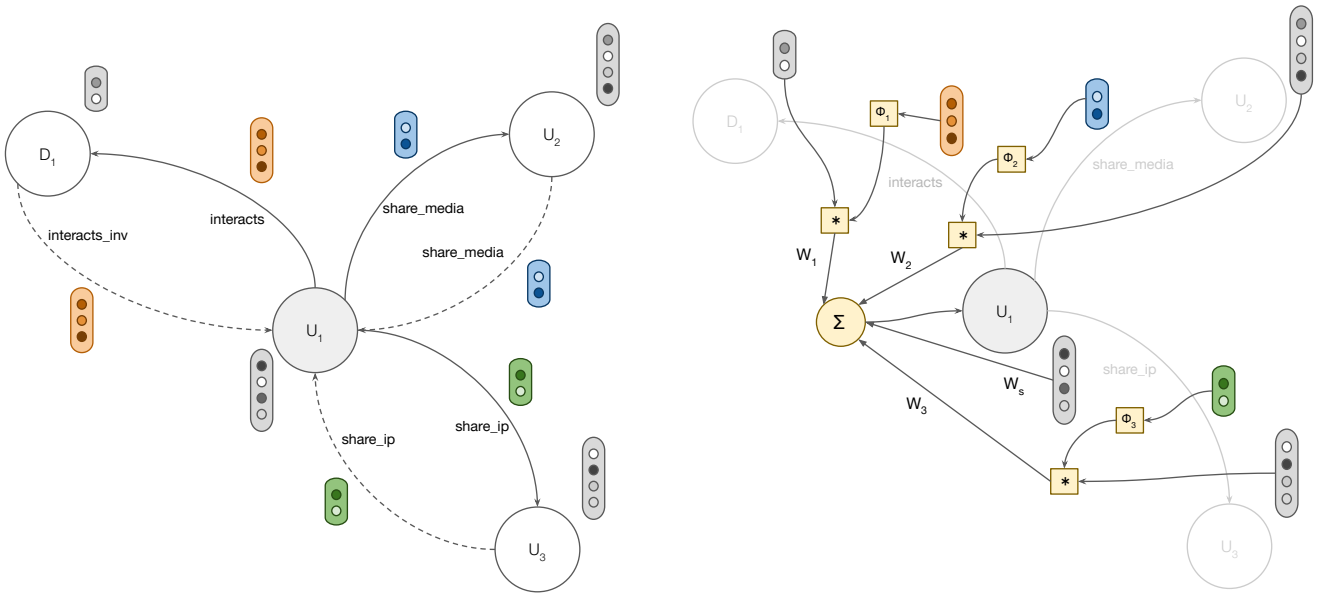
SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3531875

1 INTRODUCTION

Online platforms (Pinterest, Facebook, Snap, Youtube, etc.) are the go-to place for people to share content and information, interact, and drive influence. These platforms also give earning opportunities to users based on the engagement generated by their content. Some bad actors with malicious intent abuse these platforms, either individually or in a group, for unfair gains. Spam [30, 31] is a common form of abuse that compromises platform's integrity, performance of services like recommendation and search, and users' trust. It is an umbrella term for a wide-class of abuses: posting spam content [20]; artificial engagement boosting [19]; evading the system; diverting traffic through links; inorganic follow [5, 34]; and collusion [24]. Addressing spam by abuse-specific solutions is neither practical nor cost-effective in the industry because of their wide range and emergence of new spam vectors with product evolution. To make things worse, abusers also display adversarial nature by resorting to innovative evasive techniques against the deployed solutions. Designing a holistic solution to identify spam users (spammers), the root of the problem, yields the best strategy to effectively fight spam. Hence, we address the problem of identifying spammers on a web-scale platform in this paper.

Spammers exhibit different patterns from genuine users (non-spammers) in their profile and behavior which is evident from our analysis and other prior works [1, 21, 31]. We broadly define a user's behavior by its types and rate of activities and interaction with other users and entities (e.g., media, IP, weblinks) on the platform. This behavior can only be truly captured by a heterogeneous graph where nodes represent users and entities, edges represent the interactions of users with other users or entities, node attributes represent details of a user or an entity, and edge attributes represent interaction type and other information. Detecting spammers on web-scale platforms using such a graph is challenging because of its large size and structural complexity. Prior works [2, 18, 24, 25] for abuse detection using graphs either address a specific kind of abuse or use a smaller setting that does not scale to a large graph, or does not capture the spectrum of behavior that enables a solution with a wide abuse coverage. Recently graph neural networks (GNNs) [21, 27, 36] have been explored to detect abuse on a platform but have similar shortcomings as others or lack pragmatism. In this paper, we provide analysis on a large real dataset from Pinterest to get insights into differentiating patterns of spammers from



(a) User Interaction Graph **(b) *SEINE* update**

Figure 1: A schematic overview of *SEINE*. The left figure (a) is a subgraph of the user interaction graph showing the neighbors (users and domains) of a user U_1 with the node and edge features. The right figure (b) describes the convolution for user U_1 at a given layer in the user-interaction graph. *SEINE* first calculates edge weights w_e using a relation specific function ϕ_r and the edge features. Then it aggregates node embeddings of the neighbors using relation specific filters W_r and edge weights w_e. Finally, a sum of the neighbors aggregate and U_1's embedding is transformed to produce the updated embedding for U_1. Please refer to Section 5 for details.

non-spammers. We use these insights to design a new kind of heterogeneous graph and learn a model on such a graph to detect spammers on a web-scale platform. Our major contributions are:

- A user-entity graph that uses edge types and attributes along with node attributes to capture more behavior than considered in prior works. It also selectively transforms some of the user-entity interactions into user-user edges to reduce the size of the graph and enables learning a model on a large-scale graph.
- A graph neural network model *SEINE* (**S**pam D**E**tection using **I**nteraction **NE**tworks) that considers neighborhood along with edge types and attributes for differentiating spammers from non-spammers.
- An extensive empirical evaluation on a large graph from Pinterest which establishes that *SEINE* has **40**% better performance over strong ablation baselines. *SEINE* achieves comparable performance to the state-of-the-art methods on a public dataset while being simple and pragmatic to be used in a large-scale production system.

We discuss related work in Section 2, user interaction graph in Section 3, problem definition in Section 4, model in Section 5, empirical evaluations in Section 6, and conclusions in Section 7.

2 RELATED WORK

Researchers have proposed a variety of spam detection techniques using both content and behavior features for various domains: web [11, 30], email [4, 7, 37], microblogging [15, 34], social networks [5, 38], and reviews [13, 16]. Most of these techniques apply classical machine learning, use human-created features represented in tabular data format on a small-scale dataset, and only partially capture the complex behavior of spammers. Graph representation [2] has been explored for anomaly detection and then applied for spam. [25] proposes an unsupervised technique to find anomalous subgraphs, [18] uses density and grid-based clustering to find anomalous clusters, and [5] clusters users based on the similarity of their activities over a sustained time. Prior works have also explored mining techniques [6, 8, 14, 24, 28] for detecting anomalous groups. Most of these graph techniques either use a homogeneous graph, or relatively small graphs and can not be scaled to large graphs, or detect only clusters of colluding spammers and do not address individual spammers. Recent advances in Graph Neural Networks [12, 17, 27, 29, 32] and transformer architectures [33] have received some attention for detecting spammers. Liu et al. [21] proposes an edge-type weighted graph convolution network over a user-device graph to detect abusive accounts. [36] proposes a residual layer-GNN whereas [27] proposes a recursive and flexible neighborhood selection guided multi-relational GNN to detect evasive abusers. Both these techniques identify only a narrow set of abusers.

3 USER INTERACTION GRAPH

In this section, we first discuss an analysis of users' behavior on a real dataset from Pinterest and then describe the construction

of our graph framework based on these insights. Pinterest dataset includes a range of user activities over two months: Pin creation with weblinks, interaction with other users' content, user following, and other engagements.

3.1 Spam Behavior Analysis

Here, we present a behavioral analysis of spammers and non-spammers that uncovers distinctive patterns between them. Our analysis strongly motivates representing users' details and interactions in a graph setting to detect spammers.

We analyzed users' following ratio which is the fraction of users followed by a given user who later followed it back. We found the following ratio of spammers (0.1%) to be significantly lower than non-spammers (16%) because spammers blindly follow a large number of users who do not follow them back. This highlights differences in the behavior of individual spammers and non-spammers. Spammers also tend to collude and share resources, e.g., IP, device fingerprint, content, etc., to scale their spam activities at a controlled cost and thus, maximize gain. We analyzed IP sharing between pairs of users and observed a probability of 0.97 for both users being spammers compared to a probability of 0.027 for one of them being a spammer and only a probability of 0.001 for both users being non-spammers. We also studied content sharing between pairs of users using Jaccard's metric and observed that spammers have a much higher sharing than non-spammers.

Next, we analyzed users' interaction with weblinks using a weighted users-domain bipartite graph where the weight of an edge is the frequency of a user's interactions with a domain. We found that non-spammers interacted with twice more domains than spammers but interacted only half times that of spammers. This shows that spammers tend to highly engage with a small number of domains. We also investigated neighborhood patterns in this bipartite graph. For a user, we define the 1-hop neighbors as users who have interacted with the same domain. We found that spammers have 5 times more spammers in their neighborhood than non-spammers. We also computed the average shortest path between spammers and found it to be 1.4 times more than non-spammers. This shows that spammers tend to highly engage with a small set of disconnected domains in groups. We also analyzed user-user graphs based on IP and content sharing which revealed that spammers have 6 times and 4 times more spammers in their ego-networks than non-spammers in these graphs respectively.

3.2 Graph Construction

Here, we describe our approach to construct the graph that holistically captures users' profile details and overall behavior. We represent users as nodes and details specific to users as node attributes. We categorize entities into two groups and take a mixed approach to represent users' interactions with entities. For entities with additional information like domain, we represent these as nodes and their details as node attributes. We represent users' interactions with these entities by typed edges with attributes. Edge types distinguish users' interactions with entities by entity types. For entities with no additional information like IP, we create a typed edge with attributes between a pair of users based on their interaction with the entity type. This mixed approach simultaneously captures more behavior details while limiting the size of the graph. A schematic overview of this user interaction graph is shown in Figure 1.

4 PROBLEM DEFINITION

Formally, our graph is represented as a directed heterogeneous multi-graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{T}_v\}$ with nodes $v_i \in \mathcal{V}$ and edges $(v_i, r, v_j) \in \mathcal{E}$, where $r \in \mathcal{R}$ represents a relation and $\mathcal{T}_v$ represents the set of all node types. Nodes $\mathcal{V}$ in the graph consists of users $u \in \mathcal{U}$ and entities $\mathcal{O} = \{O^p\}$ of different types $p \in \mathcal{T}_v \setminus u$ which the users interacted with. Additionally each node v of type $t \in \mathcal{T}_v$ is associated with a set of features $x_v \in \mathbb{R}^{d_t}$ and each edge $e \in \mathcal{E}$ of relation $r \in \mathcal{R}$ is associated with features $x_e \in \mathbb{R}^{d_r}$.

Given the graph $\mathcal{G}$, our task is to learn a function $f(u, \mathcal{G}) \to \mathbb{R}$ to predict the spam probability of a user node $u \in \mathcal{U}$ using both its user features x_u and neighborhood with edge types and attributes.

5 MODEL

Here, we describe our end-to-end learned model *SEINE* which consists of two main components. First, an encoder $\mathcal{ENC}$ maps a node v to an embedding space $\mathbb{R}^d$ using its local neighborhood and features. Second, an output layer predicts the probability of spam for a user node u based on its embedding $\mathcal{ENC}(u)$.

The encoder leverages rGCN [29] as its building block due to its effectiveness in simultaneously capturing both the relational neighborhood structure and the local information associated with nodes. The encoder consists of several layers of convolution in the graph and is shown schematically in Figure 1. Given a latent representation $h_v^{(l)}$ of a node $v \in \mathcal{V}$ in the l-th layer of the neural network, the encoder can be represented as:

$$h_v^{(l+1)} = \sigma\left(W_s^{(l)} h_v^{(l)} + \sum_{r \in \mathcal{R}} AGG_r(\mathcal{N}_v^r)\right), \tag{1}$$

where σ is the RELU activation function, $W_s^{(l)}$ is the transformation matrix for self-loop and $\mathcal{N}_v^r$ is the set of neighbors of node v for relation $r \in \mathcal{R}$. $h_v^{(0)}$ is initialized by a linear projection of node features $x_v \in \mathbb{R}^{d_t}$ of node v using a node type $t \in \mathcal{T}_v$ specific transformation matrix W_t:

$$h_v^{(0)} = W_t x_v\,. \tag{2}$$

We incorporate edge features in the per-relation neighbor aggregator function AGG_r. For each edge $e = (v, r, v')$, we obtain the edge weight $w_e^{(l)} \in \mathbb{R}$ based on its features $x_e \in \mathbb{R}^{d_r}$ using the relation-specific learned function $\phi_r^{(l)} : \mathbb{R}^{d_r} \to \mathbb{R}$ as

$$w_e^{(l)} = \phi_r^{(l)}(x_e) = \sigma\left(MLP_r^{(l)}(x_e)\right), \tag{3}$$

where $MLP_r^{(l)}$ is a two-layer perceptron and σ is the logistic sigmoid function. The aggregator incorporates these edge weights to summarize the neighbors of a node v as follows,

$$AGG_r(\mathcal{N}_v^r) = \frac{1}{|\mathcal{N}_v^r|} \sum_{v' \in \mathcal{N}_v^r} W_r^{(l)}(w_e^{(l)} h_{v'}^{(l)}), \text{ where } e = (v, r, v') \tag{4}$$

where $W_r^{(l)}$ is the transformation matrix for relation $r \in \mathcal{R}$. Since the transformation matrix depends on the relation type, the encoder

propagates latent node feature information across the edges of the graph while taking the edge type into account.

Each user node has an output layer to learn spam probability $p(u)$ defined as:

$$p(u) = \sigma(Wh_u + b), \tag{5}$$

where $W \in \mathbb{R}^d$ is the weight matrix, b is the scalar bias, and σ is the logistic sigmoid function.

We train *SEINE* end-to-end using the labeled subset of user nodes $\mathcal{U}_l \subset \mathcal{U}$ with binary cross-entropy loss as follows:

$$L = -\frac{1}{|\mathcal{U}_l|} \sum_{u \in \mathcal{U}_l} y_u \log p(u)) + (1 - y_u) \log(1 - p(u)), \tag{6}$$

where y_u is the spam label for a user node u.

6 EXPERIMENTS

In this section, we present a comprehensive evaluation on a large real Pinterest dataset with multiple ablations and qualitative analysis and provide comparisons with state-of-the-art methods on a public Amazon Fraud detection dataset [9] to establish the effectiveness of *SEINE*.

6.1 Experiment Setup

Datasets. *Pinterest Dataset.* This dataset consists of user activities on Pinterest — a visual discovery engine for exploring billions of inspirations. Users on Pinterest can create, browse, search and save (repin) Pins on boards. Each Pin has media content that may link to an external web page. Labels for users are obtained by a mix of human reviews and spam users detected by highly precise production systems both of which enforce Pinterest's community guidelines [1].

For each user in the dataset, we collect their activities over a period of 2 months to get good coverage of their behavior. We time-split the data into training and test to ensure that these are non-overlapping consecutive time windows of activities. We include only labels for users who were detected and actioned after the end of the time window to replicate the actual production scenario and avoid data leakage.

Our graph, constructed as described in 3.2, includes users' interactions with three types of entities: domain (derived from the associated Pin weblinks), IP, and content. Domains are represented as nodes in the graph along with users whereas users' interactions with IPs and content are reduced to user-user edges. We drop domains that have more than 10,000 users interacting with them and users who have less than 10 interactions with domains. A user node has 15 features including locale, signup details, follow features, and frequency of changes in profile attributes. A domain node has 3 features including spamminess score and spam label. The graph consists of 4 relations, namely,

(1) U-I-D: connecting users to domains having at least 10 interactions,
(2) D-I-U: connecting domains to users having at least 10 interactions,
(3) U-E1-U: connecting users who share the same IP at least once, and
(4) U-E2-U: connecting users who share content at least once.

A user-domain edge has 35 features including the number of interactions and proportion of interactions to this domain. A user-user edge has features like frequency of sharing and Jaccard's metric over the set of IPs and content between corresponding users. Table 1 shows the number of nodes and edges for each relation in both the train and test split of the Pinterest dataset. *Amazon Fraud Dataset.* This is a subset of Amazon's product dataset [22]. We construct a heterogeneous graph as in [9] with three relations, namely,

(1) U-P-U: connecting users reviewing at least one common product,
(2) U-S-V: connecting users having at least one same star rating within one week, and
(3) U-V-U: connecting users with top 5% mutual review text similarities (measured by TF-IDF) among all users.

The number of nodes and edges for each relation is shown in Table 1. We use only 40% of the nodes for training similar to other works on this dataset.

Dataset	Node Type	#Nodes	#Labeled Nodes (% Spam)	Relation	#Edges
Pinterest-Train	User	22.21M	85,626 (63%)	U-I-D	76.18M
	Domain	2.37M	-	D-I-U	76.18M
	ALL	24.58M	-	U-E1-U	801.01M
				U-E2-U	265.14M
				ALL	1.22B
Pinterest-Test	User	21.19M	42,211 (63%)	U-I-D	67.71M
	Domain	2.52M	-	D-I-U	67.71M
				U-E1-U	923.99M
				U-E2-U	341.32M
				ALL	1.40B
Amazon-Fraud	User	11,944	11,944 (9.5%)	U-P-U	175,608
				U-S-U	3,566,479
				U-V-U	1,036,737
				ALL	4,398,392

Table 1: Statistics of nodes and edges per relation for various graph datasets used for empirical evaluations. Column 4 shows the number of labeled user nodes in each dataset along with the percentage of spammers among them.

Baselines. For ablation evaluations, we use a no-graph model using only user features and a *SEINE* variant without edge features to show the importance of incorporating interactions and edge features in the model. For comparison with the state-of-the-art methods, we use RLC-GNN [36] and RioGNN [27] (a multi-layer extension of the CARE-GNN [9] model) both of which leverage reinforcement learning (RL) to select an optimal number of neighbors to address spammer evasiveness.

Experiment Settings. Based on multiple hyperprameter tuning runs, we select node embedding size (d) of 256, batch size of 512, 2 network layers, a learning rate of 9.5×10^5, LayerNorm [3] after each GCN layer, and L2 regularization weight of 0.0001 for all models. To improve the training efficiency on a large-scale graph, we employ neighbor sampling [12] with a fanout factor of 50 for each relation for both layers to train all graph neural network models.

[1] https://policy.pinterest.com/en/community-guidelines

Model	Recall@FPR1(%)	ROCAUC@FPR1(%)
SEINE	80.06	69.52
SEINE w/o edge features	77.95	64.85
User only baseline	57.31	33.41

Table 2: Performance of *SEINE* and baseline methods on the Pinterest test dataset.

Model	Recall@FPR1(%)	ROCAUC(%)
SEINE	73.84	96.69
RLC-GNN [36]	-	97.48
RioGNN [27]	-	96.19

Table 3: Comparison of *SEINE* with the state-of-the-art methods on Amazon Fraud dataset.

We implement all models in Pytorch [26] using DGL [35] and train them on 2 GPUs with Adam optimizer. We train all models for a maximum of 2,000 steps and use the validation set (a subset of nodes in the training graph) for early stopping.

Evaluation Metrics. A key requirement of any spam detection system is to have a low *false-positive rate (FPR)* while having a high recall in identifying spammers to ensure minimal impact on genuine users. Hence we use recall at an FPR of 1% as our evaluation criteria to compare different models. We also report the normalized area under the ROC curve truncated at 1% FPR to analyze models' performance in regions with low-false positive rates.

6.2 Results

We report test set Recall@FPR1 and AUROC@FPR1 for *SEINE* and other baselines on Pinterest dataset in Table 2. Our model outperforms all baselines by achieving 80.06% Recall@FPR1 and 69.52% AUROC@FPR1. *SEINE* has 40% higher recall over no-graph user features only baseline demonstrating the effectiveness of graph in identifying spam behavior. We also see that learning edge weights from attributes help *SEINE* get 3% improvement in recall at 1% FPR over its variant with no edge features.

We show comparisons with the state-of-the-art methods on the Amazon dataset in Table 3. *SEINE* outperforms *RioGNN* and has comparative performance with *RLC-GNN* despite being much simpler in architecture and not using expansive RL techniques for neighbor selection. Both *RioGNN* and *RLC-GNN* compute similarity of a node with all its neighbors to perform neighbor selection and hence cannot scale to web-scale graphs. This shows the superiority of *SEINE* in terms of scalability and performance over state-of-the-art methods in learning a good representation of users by exploiting their rich interactions on the platform.

To further understand the learned representations of the users, we analyzed embeddings of spammers identified by *SEINE*. We use DBSCAN [10] to cluster the users by their embeddings and use UMAP [23] to visualize the clusters as shown in Figure 2. We sampled a few users from each of the big clusters and analyzed their spam behavior. We observed that each cluster corresponds to a different abusive behavior as annotated in the figure. Further, users exhibiting similar abuse behavior are closer to each other in this space. This shows that *SEINE* can extract a wide range of abusive behavior and learn meaningful representations of them.

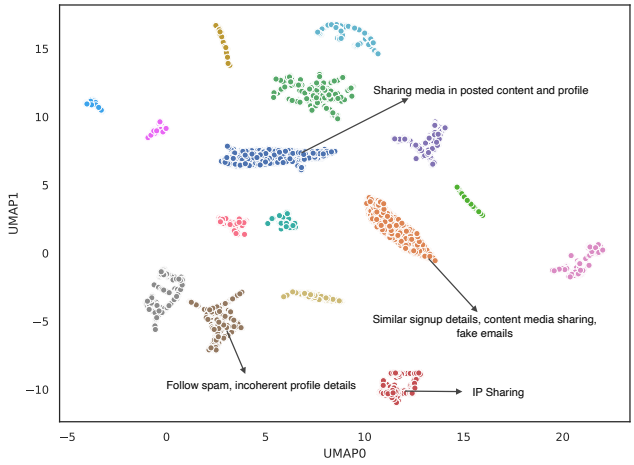


Figure 2: UMAP visualization of clusters of users detected as spam by *SEINE*. Some clusters are annotated with the abusive behavior exhibited by the users in the cluster. Users exhibiting similar abuse behavior are closer to each other in the embedding space demonstrating that *SEINE* can extract a wide range of abusive behavior and learn meaningful representations of them.

7 CONCLUSION

Addressing spam on a web-scale platform is very challenging because of its scale and a wide range of abusive behavior of spammers. In this paper, we proposed a new heterogeneous graph framework that holistically captures users' behavior and facilities learning a powerful graph convolution network model to identify spammers with high coverage. Our model considers additional graph structures like edge types and attributes for learning which boosts its performance. We provide strong empirical results on two real datasets to show its superior performance over alternative and ablation methods. Our results also show that the method achieves a very high recall with a small false-positive rate and hence, can be deployed in real production to effectively address spam.

ACKNOWLEDGMENTS

We thank Rundong Liu, Omkar Panhalkar, Yuanfang Song, and Dennis Horte for their valuable inputs. We also thank Maisy Samuelson for her review and feedback.

REFERENCES

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. OddBall: Spotting Anomalies in Weighted Graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 410–421.
[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph Based Anomaly Detection and Description: A Survey. *Data Mining and Knowledge Discovery* 29, 3 (2015), 626–688.
[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450* (2016).
[4] Enrico Blanzieri and Anton Bryl. 2008. A Survey of Learning-Based Techniques of Email Spam Filtering. *Artificial Intelligence Review* 29 (03 2008), 335–455.

[5] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering Large Groups of Active Malicious Accounts in Online Social Networks. In *ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 477–488.

[6] Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. 2004. Fully Automatic Cross-Associations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 79–88.

[7] Gordon V. Cormack. 2007. Email Spam Filtering: A Systematic Review. *Foundations and Trends in Information Retrieval* 1, 4 (2007), 335–455.

[8] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. 2003. Information-Theoretic Co-Clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 89–98.

[9] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing Graph Neural Network-Based Fraud Detectors against Camouflaged Fraudsters. In *ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, 315–324.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 226–231.

[11] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. 2004. Combating Web Spam with Trustrank. In *International Conference on Very Large Data Bases*. VLDB Endowment, 576–587.

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *International Conference on Neural Information Processing Systems*. Curran Associates Inc., 1025–1035.

[13] Atefeh Heydari, Mohammad ali Tavakoli, Naomie Salim, and Zahra Heydari. 2015. Detection of Review Spam: A Survey. *Expert Systems with Applications* 42, 7 (may 2015), 3634–3642.

[14] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. FRAUDAR: Bounding Graph Fraud in the Face of Camouflage. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 895–904.

[15] Xia Hu, Jiliang Tang, Yanchao Zhang, and Huan Liu. 2013. Social Spammer Detection in Microblogging. In *International Joint Conference on Artificial Intelligence*. AAAI Press, 2633–2639.

[16] Nitin Jindal and Bing Liu. 2007. Review Spam Detection. In *International Conference on World Wide Web*. Association for Computing Machinery, 1189–1190.

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[18] Kingsly Leung and Christopher Leckie. 2005. Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters. In *Australasian Conference on Computer Science*. Australian Computer Society, Inc., 333–342.

[19] Yixuan Li, Oscar Martinez, Xing Chen, Yi Li, and John E. Hopcroft. 2016. In a World That Counts: Clustering and Detecting Fake Social Engagement at Scale. In *International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 111–120.

[20] Chengfeng Lin, Jianhua He, Yi Zhou, Xiaokang Yang, Kai Chen, and Li Song. 2013. Analysis and Identification of Spamming Behaviors in Sina Weibo Microblog. In *Workshop on Social Network Mining and Analysis*. Association for Computing Machinery, Article 5, 9 pages.

[21] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous Graph Neural Networks for Malicious Account Detection. In *ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 2077–2085.

[22] Julian John McAuley and Jure Leskovec. 2013. From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews. In *International Conference on World Wide Web*. Association for Computing Machinery, 897–908.

[23] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426* (2018).

[24] Hamed Nilforoshan and Neil Shah. 2019. SliceNDice: Mining Suspicious Multi-Attribute Entity Groups with Multi-View Graphs. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE.

[25] Caleb C. Noble and Diane J. Cook. 2003. Graph-Based Anomaly Detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 631–636.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 8024–8035.

[27] Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S. Yu. 2021. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2021).

[28] B. Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju, and Christos Faloutsos. 2009. EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs. In *IEEE International Conference on Data Mining Workshops*.

[29] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *European Semantic Web Conference*. 593–607.

[30] Nikita Spirin and Jiawei Han. 2012. Survey on Web Spam Detection: Principles and Algorithms. *ACM SIGKDD Explorations Newsletter* 13, 2 (2012), 50–64.

[31] Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. 2012. Spammer Behavior Analysis and Detection in User Generated Content on Social Networks. In *International Conference on Distributed Computing Systems*.

[32] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *International Conference on Neural Information Processing Systems*. Curran Associates Inc., 6000–6010.

[34] Alex Hai Wang. 2010. Don't Follow Me: Spam Detection in Twitter. In *International Conference on Security and Cryptography (SECRYPT)*. IEEE.

[35] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315* (2019).

[36] Yufan Zeng and Jiashan Tang. 2021. RLC-GNN: An Improved Deep Architecture for Spatial-Based Graph Neural Network with Application to Fraud Detection. *Applied Sciences* 11 (06 2021), 5656.

[37] Yao Zhao, Yinglian Xie, Fang Yu, Qifa Ke, Yuan Yu, Yan Chen, and Eliot Gillum. 2009. BotGraph: Large Scale Spamming Botnet Detection. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX Association, 321–334.

[38] Yin Zhu, Xiao Wang, Erheng Zhong, Nanthan N. Liu, He Li, and Qiang Yang. 2012. Discovering Spammers in Social Networks. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 171–177.