Graph-Aware Deep Fusion Networks for Online Spam Review Detection

Li He*, Guandong Xu*, Shoaib Jameel[b], Xianzhi Wang[a], Hongxu Chen[a]
[a]University of Technology Sydney, Sydney, Australia
[b]University of Southampton, Southampton, United Kingdom
{li.he-1, guandong.xu, xianzhi.wang, hongxu.chen}@uts.edu.au, {M.S.Jameel}@southampton.ac.uk
* Corresponding authors

***Abstract*—Product reviews on e-commerce platforms play a critical role in shaping users' purchasing decisions. Unfortunately, online reviews sometimes can be intentionally misleading to manipulate the ecosystem. To date, existing methods to automatically detect "spam reviews" either focus on sophisticated feature engineering with traditional classification models or rely on tuning neural networks with aggregated features. In this paper, we develop a novel graph-based model, namely Graph-aware Deep Fusion Networks (GDFN) that utilizes information from relevant metadata (review text, features of users, and items) and relational data (network) to capture the semantic information from their complex heterogeneous interactions via graph convolutional networks. Besides, GDFN also uses a novel fusion technique to synthesize low and high-order interactions with propagated information across multiple review-related subgraphs. Extensive experiments on publicly available datasets show that our proposed model is effective and outperforms several strong state-of-the-art baselines.**

***Index Terms*—E-commerce, Online Review, Spam Detection, Graph Convolutional Networks**

I. Introduction

E-COMMERCE companies such as Amazon and eBay have earned approximately \$3.5 trillion in sales in 2019 and are anticipating an increase to \$4.9 trillion by the end of 2021, according to Shopify.com. Online e-commerce has demonstrated unique importance during the COVID-19 pandemic and enabled hundreds of millions of consumers to purchase products anytime and anywhere around the world. Currently, customers can also share their shopping experiences by rating items, writing reviews, and answering questions related to the products that they have used in the past or recently purchased online.

Online reviews play an important role in e-commerce as they impact the purchasing decisions of approximately 93% of people, according to Ingniyte.co.uk. Unfortunately, online reviews can be deliberately injected (a.k.a., "spam reviews") to mislead potential customers [1] for various unethical reasons, such as unfair marketing or online brand attacks [2]. According to BrightLocal.com, 74% of consumers in 2019 have encountered spam reviews yet failed to recognize them. It has thus become very crucial to devise effective methods that can identify spam reviews automatically so that these platforms remain reliable [3].

Despite various efforts on automatic spam review detection, most of them largely rely on learning from engineered features and lack generalizability. For example, traditional statistical learning methods usually use supervised classifiers, e.g., support vector machines [4] (SVM), logistic regression [1], and Naïve Bayes [5], to detect unusual patterns based on extracting review-specific semantic information [6]. Such feature-centric methods usually ignore correlations among reviews, users, and items. As shown in Figure 1(a), reviews on Yelp are useful and can be used as a reliable guide for users to make a choice. However, experience tells us that only looking at the review may mislead us into making an unwise decision, and we may need to double-check the information (e.g., credibility, tastes, biases, and beliefs) about reviewers. Similarly, only leveraging review text as features can be problematic as they are sometimes ambiguous, and credibility cannot be guaranteed at all times.

To address the limitations of existing methods, we hypothesize that modeling the information gathered from reviews, users, and items could help substantially improve the performance and generalizability of online spam review detection systems. We thus develop a novel model **Graph-aware Deep Fusion Networks** (**GDFN**), which is capable of capturing the heterogeneity and complex interactions among different features obtained from users, and their reviews on the items. GDFN considers the user-review-item network to formulate the problem as a graph-based classification task, in which reviews are labeled as spam or non-spam. At the local feature space level, GDFN can distill the graph's structural information from different types of features (i.e., user-item bipartite graph, review text graph, user-review graph, and item-review graph). For example, GDFN extracts structured information networks from the original unstructured textual information of review data, which is potentially helpful for learning strong discriminative features in spam review detection. At the global level, GDFN can also learn the macro view of the heterogeneous information network that is aggregated from the extracted individual graphs. GDFN then learns reinforced cross-graph features that depict the useful correlations among all available metadata under a unified framework to detect spam reviews.

Existing methods either concatenate multiple vectors or use selected pooling methods to obtain a fixed dimensional

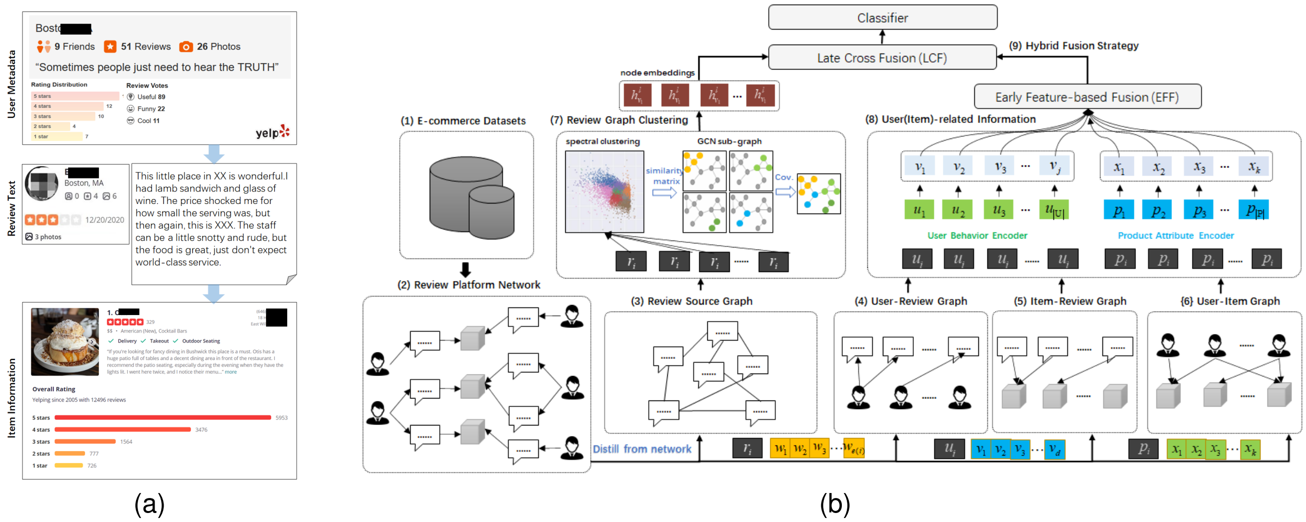


Fig. 1. **(a):** Yelp Review Platform: the user metadata page (top) and the item information page (bottom). The comment area (middle) with rating and raw review text, then the post review can be treated as a bridge between user and item. **(b)**: Our proposed framework comprises of the following parts: (1) Training datasets from e-commerce platforms. (2) Heterogeneous Network from review platforms. (3)-(6) Review source homogeneous graph and User-Review/Item-Review/User-Item bipartite graph distilled from component (2). (7) Review Graph Clustering. (8) User (Item)-related feature embeddings. (9) Hybrid Fusion Strategy.

vector. The limitation of such methods is that they may result in information loss, especially under heterogeneous feature scenarios. As a result, we propose a new fusion method to allow flexible information exchange and the interplay between different local views of graph structural information. Instead of applying concatenation of embeddings of various graph views, we adopt the outer product between subgraph-specific embeddings to obtain the fused features. The reason is that the outer product kernel outputs an N-way tensor that favors the strong expressiveness of both lower and higher-order feature interactions. When modeling the above information together, there are several underlying challenges. For instance, a usual representation learning approach is not universal to different graph structures distilled from distinct features. Besides, a single general graph convolutional network (GCN) is not adequate to capture the unique characteristics of different graphs constructed from multiple feature spaces in a complex heterogeneous environment of online review platforms.

In summary, our key contributions are as follows:

- We propose a novel GCN-based heterogeneous graph-aware spam review detection framework that is more expressive than existing text-based methods as it seamlessly captures relevant metadata and relational data to strengthen the review embedding for the underlying task.
- We exploit unsupervised approaches to learn the constructed review graph, which effectively resolves the problem of lack of labeled data. We also develop a novel fusion strategy to model multiple types of interaction information effectively.
- Extensive experiments with large-scale reviews from two real-world datasets demonstrate that our framework achieves consistent improvements over state-of-the-art methods. Our ablation study demonstrates the effectiveness of novel components of GDFN.

II. Related Work

A. Feature-centric Methods

Traditional statistical methods rely on extracting different features from textual reviews, followed by learning a language model. In [1], the authors first identified three types of spam reviews, which are untruthful opinions, reviews on brands only, and non-reviews, and then analyzed real-world datasets from Amazon. They extracted review-centric, reviewer-centric, and product-centric features, and used them as input to a logistic regression (LR) model. Recently, in [7], the authors summarize eleven platform-independent features from the word level, the semantic level, and the structural level to discriminate between fraud and normal items. They used Xgboost as a binary classifier, and their evaluation results indicated that CATS achieves both high precision and recall. In [4], the authors approached the problem using three strategies as features in Naïve Bayes and SVM classifier. The authors in [8] attempted to use Long Short-Term Memory (LSTM) framework to detect spam reviews. They established three types of layers to predict spam reviews, the input layer for receiving data, the hidden layer of LSTM, and the output layer, respectively.

B. Graph-based Methods

Graph-based methods have been popularly applied to capture text features among different entities. The first Graph Neural Network [9] (GNN)-based spam review detection method was proposed by [10], who built a heterogeneous "review graph" to represent the relationship among reviewers, reviews, and online sellers. In [11], the authors utilized spam features for modeling review datasets as heterogeneous information networks to map spam review detection procedure into a classification problem in such networks. In the classification step, they proposed meta-path concepts to find feature importance and calculate the weight. The authors in

TABLE I
NOTATIONS

Notation	Description
R	The set of review source, $\{r_1, r_2, \cdots, r_{\|R\|}\}$
U	The set of users, $\{u_1, u_2, \cdots, u_{\|U\|}\}$
P	The set of items, $\{p_1, p_2, \cdots, p_{\|P\|}\}$
r_i	e_i words $\{w_1^i, w_2^i, \cdots, w_{e_i}^i\}$,$r_i \in R$
Y_i	The tuple formula, denoted as $\{u_j, r_i, p_k\}$
$\mathcal{G}_i$	A undirected graph of each review cluster
$< V_i, E >$	The node and edge set of $\mathcal{G}_i$
y_i	The ground-truth label, $y_i \in \{Y, N\}$
$\mathcal{T}$	The outcome fusion tensor
$f(.)$	The classifier function

[12] presented a neural network-based graph model, named Graph Embeddings for Malicious accounts (GEM), which both considered "device aggregation" and "activity aggregation" in heterogeneous graphs. So far, these methods have focused on shallow encoders, i.e., matrix factorization. There is no parameter sharing, and every node has its unique embedding vector and the inherent "transductive" features are impossible to generate embeddings for unseen nodes during training and do not incorporate node features.

Recent years have witnessed a growing interest in utilizing the "message-passing" methodology in graphs [13], which learns how to aggregate information from each type of neighbor using Markov Random Field (MRF) techniques implicitly. In [14], the authors presented the GraphSAGE model, which achieves significant improvements compared with previous methods such as DeepWalk [15] and SemiGCN [16]. This method overcomes the limitation of applying GCN in transductive settings with a specified Laplacian matrix. A model-based Graph Convolutional Neutral Networks (GCNN) for spam-bot detection is proposed in [17], which proposed an inductive representation learning approach for spam review detection based on the reviewer profile information and the social network graph on Twitter datasets, and the inductive representation learning method used in their approach is similar to that of GraphSAGE. In short, GCN-based methods have been applied in various domains, such as spam advertisement identification [18], recommendation system [19]–[21], social spammer detection [22], rumor detection [23] and so no.

The above methods depend only on the local information of surrounding neighborhoods of a target node, making the model sometimes noisy and thus ineffective. The multiple convolutional layers may cause an over-fitting and over-smoothing problem. To overcome the limitations inherent in existing methods, we design a novel model that exploits review-user-item three-fold information and distillsreview clustering and user-item level information.

C. Preliminaries

In this section, we introduce some fundamental concepts that are necessary to understand our model. The notations used in this paper are summarized in Table 1.

1) Graph Convolutional Networks (GCN): Recently, there is an increasing interest in utilizing convolutions in graph-based methods. GCN is one of the most effective graph-aware models, whose convolution operation is considered as a general layer-wise propagation architecture as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{A}}^{(l)}\mathbf{W}^{(l)}) \tag{1}$$

The input is an adjacency matrix $\mathbf{A}$ and a feature matrix $\mathbf{W} \in \mathbb{R}^{N \times E}$, where $\tilde{\mathbf{A}} = \tilde{D}^{-\frac{1}{2}}\mathbf{A}\tilde{D}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix of graph $\mathcal{G}$ with added self-connections and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. σ is a non-linear activation function, such as the ReLU(.) = max(0, ·). In [16], a propagation structure is proposed that can be separated into two components: aggregation and combination. In general, for a GCN with L layer, aggregation and combination sub-layers at l^{th} layer $(l = 1, \cdots, L)$ can be written as:

$$\mathbf{H}_{N(v)}^{(l)} = \sigma\left(\mathbf{W}^{(l)} \cdot AGG\left(\left\{\mathbf{H}_v^{(l-1)}, \forall v \in \mathbf{N}(v)\right\}\right)\right) \tag{2}$$

$$\mathbf{H}_v^{(l)} = CONCAT\left(\mathbf{H}_v^{(l-1)}, \mathbf{H}_{\mathbf{N}(v)}^{(l)}\right) \tag{3}$$

where $\mathbf{N}(v)$ is a set of nodes adjacent to $\mathbf{v}$, $AGG(.)$ is a function used for aggregating embeddings from neighbors of node $\mathbf{v}$. This function can be customized for specific models, e.g., mean aggregator, LSTM aggregator and pooling aggregator. The notation $\mathbf{H}_{\mathbf{N}(v)}^{(l)}$ denotes the aggregated feature of node $\mathbf{v}$'s neighborhood at l^{th} layer. $CONCAT(.)$ function is used to combine self embedding and the aggregated embeddings of neighbors, which is also a customized setup for different graph models, e.g., concatenation as in GraphSAGE [14].

2) Graph-based Clustering: Inspired by graph-based clustering approaches, we use relationships from graphs, such as the spectral clustering technique [24] to transform the data into a weighted, undirected graph based on pairwise similarities. The graph clustering methods generally build k-means graphs with unlabeled data D_u as input and extract features $F(D_u)$. With these features, they find k-means for each sample D_u using cosine similarity. We develop two different versions of k-means graphs, which are:

- The relationship R between the two nodes. Intuitively, it can be understood as whether two nodes are neighbors in the view of each k-means graph.

$$R_{C_i}^{(n_0,n_1)} = \begin{cases} 1 & \text{if } (n_0, n_1) \in \mathcal{E}\left(\mathcal{G}_{c_i}\right),\ i = 1, 2, \ldots, N, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

  where $\mathcal{G}_{c_i}$ denote to the k-means graph of i-th clustering, and $\mathcal{E}$ denotes all edges of a graph. Here, n_0 and n_1 represent two nodes in the graph.
- The affinity $\mathbf{M}$ is defined as the Euclidean Distance (denoted as Dist) measured in the feature space,

$$\mathbf{M}_{C_i}^{(n_0,n_1)} = \text{Dist}\left(\langle F_{C_i}(n_0), F_{C_i}(n_1)\rangle\right), i = 1, 2, \ldots N \tag{5}$$

  Here, n_0 and n_1 are connected by the affinity vector $\mathbf{M}_{C_i}$ in $\mathcal{C}_i$ clustering graph.

III. METHODOLOGY

In a nutshell, in our model, well-tailored representation learning models for each sub-graph are elaborated to preserve the uniqueness of the derived features. We first utilize spectral clustering to build unsupervised learning modules for learning the review data similarity matrix. A multi-layer convolutional neural network is constructed to capture information from similar neighborhoods of a node, where the convolutions are defined on a graph structure. We then employ a hybrid fusion strategy [25] to obtain discrete values from user behavior and item attribute information. Specifically, we first adopt the idea of "early feature-level fusion" to exploit latent relation among attributes, then apply a "late cross fusion" method to exploit the correlation and interaction among processed modalities.

A. Model Overview

An online review instance is defined as an ensemble representing three types of information $A = \{R, U, P\}$, where R is a set of review text, U is the user metadata and profiles, and P is the corresponding item attributes. By leveraging multi-level features to obtain a fusion tensor $\mathcal{T}$, we build a classifier to learn the mapping relation from input tensor to output prediction labels. Our novel model Graph-aware Deep Fusion Networks (GDFN), as illustrated in Figure 1(b), automatically predicts spam reviews based on a unimodal graph to cluster similar review texts for extracting aggregation-based semantic features. We then encode user (item)-level information to strengthen the final tensor representation. By modeling each level of information in A using relatively independent processes, the output of each encoder becomes the specific individual embeddings. The graph-aware representation learns semantic correlations from the cluster network and aggregates neighbor information from multiple subgraphs. The fusion module is to explicitly model interactions among reviews, users and items, denoted by fusion tensor $\mathcal{T}$, including three types of combinations: shallow-level (review text only), medium-level (two-dimensional matrix) and top-level (three-dimensional tensor). The fused tensor is fed into fully-connected layers with a softmax layer to perform review classification.

B. Graph-aware Representation

Our goal is to learn a novel graph to model the interaction among similar review source r_i from individuals u_j and apply it to a different item p_k. Our motivation is that some correlations between reviews with particular semantics can reveal the possibility that the source review is spam.

To achieve our objective, a graph $\mathcal{G}_{c_i} = (V_i, E_i)$ is constructed for depicting different review sets with same content i.e., V_i, where E_i is the corresponding edge set. To unify the review text input, the given source review is represented by a word-level encoder. The input is the embedding of each word in review text r_i. Due to the difference in length of each review, we perform zero-padding, appending to the tail by setting a fixed length l. Since the edge set among reviews is unknown, we consider a graph-based clustering algorithm to generate relationship R by connecting comments with similar contents. We depict this in the following two equations:

$$\forall e_{\alpha\beta} \in \mathcal{E}\left(\mathcal{G}_{C_i}\right), v_\alpha \in R_i, v_\beta \in R_i. \tag{6}$$

and,

$$v_\alpha \neq v_\beta, \left|\varepsilon\left(C_i\right)\right| = \frac{k \cdot (k-1)}{2} \tag{7}$$

where α and β denote two linked nodes, k denotes the number of v in $\mathcal{G}_{c_i}$ graph, and $\mathcal{E}$ and ε used to denote the graph edge sets. Let the affinity $\mathbf{M}$ incorporate the similarity between review node embeddings given by the following equation:

$$\mathbf{M}_{C_i}^{(v_\alpha, v_\beta)} = \mathrm{D}\left(\left\{r_\alpha \mid w_1^\alpha, w_2^\alpha, \cdots, w_{e(i)}^\alpha\right\}, \left\{r_\beta \mid w_1^\beta, w_2^\beta, \cdots, w_{e(i)}^\beta\right\}\right) \tag{8}$$

where r_α and r_β are seen as embedding vectors of each review text sequence and D denotes the vector distance. We use matrix $\mathbf{R} = [w_{v\alpha, v\beta}] \in \mathbb{R}^{n \times n}$ to represent the relationship between any pair of nodes v_α and v_β in graph $\mathcal{G}_{c_i}$.

After the clustering operation, the propagation features are obtained by GCN-based methods. As mentioned above, GCN can capture information from a node's one-hop and multi-hop neighbors through stacking layer-wise convolution. Given the matrix $\mathbf{R}$ depicting the matrix of relationship for review nodes in graph $\mathcal{G}_{c_i}$, the new d-dimensional node feature matrix $\mathbf{H}_l \in \mathbb{R}^{n \times d}$ represents the output clustering review embeddings: $\mathbf{X}_r$.

$$\mathbf{H}_{\mathbf{N}(v)}^{(l)} = \sigma\left(\tilde{A}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)} \cdot \mathbf{R}_{C_i}^{\mathbf{N}(v)}\right) \tag{9}$$

where l is the layer number, $\mathbf{W}^{(l)}$ is a trainable matrix shared among all nodes at layer l. We then choose to stack two sub-layers to derive the propagation learning representation denoted $AGG(.)$ and $CONCAT(.)$. An edge is associated with relationship R and the hidden state is updated as the concatenation of previous hidden states of the two nodes it links to. As a result, the $AGG(.)$ function can be written as:

$$\mathbf{h}_{\mathbf{N}(v)}^{(l)} \leftarrow AGG_{(l)}\left(\left\{\mathbf{h}_r^{(l-1)}, \forall r \in \mathbf{N}(v) \mid \mathbf{R}_{C_i}^{\mathbf{N}(v)} = 1\right\}\right) \tag{10}$$

After aggregating the neighbors' information, we follow a combination strategy described in [14] for the homogeneous graph as shown below:

$$\mathbf{h}_v^{(l)} \leftarrow \sigma\left(\mathbf{W}^{(l)} \cdot CONCAT\left(\mathbf{h}_v^{(l-1)}, \mathbf{h}_{\mathbf{N}(v)}^{(l)}\right)\right) \tag{11}$$

C. User (Item)-related Information

User (item)-related information has been popularly used in the past [11], [26], [27], where crucial social characteristic features have been used with faithful performance. For example, given more metadata and attributes about the user and item level, the model will focus on the balanced arbitration if posts are with positive or negative emotions. We extract three types of objective features including account-based features and transduction-based features.

a) User Metadata: To depict user behavior features, we use their metadata and profiles and define a feature vector $\mathbf{v}_j$ for each user $\mathbf{u}_j$. These features have also been used in [26]. These features are:

1) name of u_j registered on the website
2) date when u_j joined, the number of u_j's friends
3) number of times u_j has posted reviews
4) number of cool/funny/useful review posted by u_j
5) location of u_j

Each user feature vector $\mathbf{v}_j \in \mathbb{R}^d$ is generated, where d is the number of features. It is known that users' behavior is crucial in detecting spam reviews, e.g., the average rating given by reviewer, the standard deviation in rating and a feature indicating whether the reviewer always gave only good, average or low rating [28].

b) Item Attributes: To exploit item level features, we collect abundant attribute relativity information from an online review website (i.e., Yelp) to identify item vector $\mathbf{x}_k$. The collected attributes are listed as follows:

- number of reviews written for p_k
- average rating deviation of p_k
- which categories p_k belongs to
- location of p_k
- ratio of positive reviews against negative reviews on p_k

For each item attribute, we map all discrete values into the Gaussian space and represent them as the vector $\mathbf{x}_k \in \mathbb{R}^m$ based on the three-sigma rule to avoid the sparsity problem [29], [30].

c) Transduction: We consider attributes related to the transductive pattern of datasets, such as the average number of comments or words. In most cases, spam reviews are propagated in several fixed patterns [28]. Therefore, we use some useful data, such as the average length of all reviews posted by u_j or the average sentiment score of each p_k. Eventually, we utilize the strategy of decision-based operation [31] that unimodal feature portions will be more predictive by a pre-trained model, as it can project the raw features into a specialized embedding space [31]. To extract the unique information from individual raw data fields, we employ Factorization Machine (FM) [32] to tackle the problem of sparse data. As a result, the latent relevance among varying user-item behavior and attributes is encoded in the embedding vector with linear complexity.

D. Fusion Module

Existing works on multiple embeddings research use concatenation as fusion [18], resulting in suboptimal interactions. To tackle multiple types of interactions effectively, we utilize an fusion process that transforms the input representations into a heterogeneous tensor [33]. We use three unimodal information vectors denoted as $\mathbf{X}_r$, $\mathbf{X}_u$ and $\mathbf{X}_p$, according to the encoded representations $\mathbf{H}$, $\mathbf{v}_j$ and $\mathbf{x}_k$, respectively. Each vector $\mathbf{X}$ is augmented with an additional feature of constant values equal to 1, denoted as $\mathbf{X} = (\mathbf{X}, 1)^T$. The augmented matrix $\mathbf{X}$ is then projected into a multiple dimensional latent vector space by a parameter matrix $\mathbf{W}$, denoted as $\mathbf{W}^T \mathbf{X}_m$. Therefore, each possible multiple feature interaction among review-user-item is computed via outer product, $\mathcal{T} = f\left(\mathbf{W}^T \cdot \tilde{\mathbf{X}}_m\right)$, expressed as:

$$\mathcal{T} = \mathbf{W}^T \cdot (\mathbf{X}_r \otimes \mathbf{X}_u \otimes \mathbf{X}_p) \tag{12}$$

where $\otimes$ denotes the outer product, $\tilde{\mathbf{X}}$ is the input representation from review, user, and item level. It is a three-fold heterogeneous tensor, modeling all possible interrelations, i.e., review graph-aware aggregation features $\mathbf{H}$, and user-item interaction outcome $\mathbf{X}_u$ and $\mathbf{X}_p$. These operations result in following two benefits:

- different from simple concatenation, making use of feature vector among multiple vectors enables learning the different impacts of elements in different modalities
- reducing the dimensionality by compressing the fusion feature along with at least three directions.

E. Classification Model

We have obtained the graph-aware representation from review clustering networks, user-level behavior, and item-level attribute embeddings. Each review with these modalities can be represented as a heterogeneous tensor $\mathcal{T}$ with multiple sets. One of the advantages of the fusion model $\mathcal{T}$ is that it can tackle the missing information problem in the absence of one or two modalities. We use the heterogeneous tensor $\mathcal{T}$ as feature to detect spam reviews. The fully connected layers are applied over $\mathcal{T}$, and the $Softmax(.)$ function is used to convert the output values into probabilities which is commonly done in the literature.

$$\hat{y} = \mathrm{Softmax}\left(\mathrm{Fusion}\left(\mathbf{H}^k_{C_i}, \mathbf{X}_u, \mathbf{X}_p\right)\right) \tag{13}$$

where $\hat{y} \in \mathbb{R}^{1\times c}$ is the vector of probabilities for all the classes used to predict the labels of the reviews. Here we apply two-class for our detection task. We then train all the parameters in the GDFN models by choosing the cross-entropy loss as the objective function to optimize the classification task. The overall loss is the weighted sum of classification loss.

F. GDFN Algorithm

We provide a detailed description of GDFN approach in Algorithm 1.

IV. Experiments and Results

In this section, we evaluate the performance of our proposed GDFN model and compare our model with different strong comparative methods. By conducting ablation study, we demonstrate the preformance of the key components of our model.

A. Datasets

We evaluate our proposed method on two benchmark publicly available datasets. They are:

1) **Yelp** from [26] (Table II), which contains three public spam review datasets crawled from the Yelp website: YelpChi, YelpNYC, and YelpZip. The dataset comprises

Algorithm 1 GDFN training Algorithm

Data: Review Source r_i, User metadata u_j and Item attributes p_k

Result: Prediction Label $\hat{y}$ (Train a fixed number of epochs on the initial labeled and unlabeled sets R,U and P)

```
for each stage k do
    Step 1: Review Deep Clustering
    Execute K-means and Laplacian calculation based on review source word embedding r_i and obtain affinity matrix M, relation R and clustering graph G.
    Step 2: Graph Convolutional Networks
    Compute each review node v of each review cluster C_i in unlabeled data.
    for each cluster C of unlabeled set do
        AGG_(l)({h_r^(l-1), ∀r ∈ N(v) | R_{C_i}^{N(v)} = 1})
        W^(l) · CONCAT(h_v^(l-1), h_{N(v)}^(l))
    end for
    Step 3: User(Item) Information
    Compute user metadata vector u_j
    Compute user metadata vector p_k
    Early Feature-level Fusion: {u_j, p_k} → {x_j, x_k}
    Step 4: Late Cross Fusion
    Calculate Fusion Tensor:
    for each r_i of all cluster data do
        T_i = W^T · (x_r ⊗ x_u ⊗ x_p)
    end for
    Step 5: Classification
    Train a fixed number of epochs on the labeled spam review datasets R.
end for
return Prediction label and Accuracy based on Tensor T
```

TABLE II
DATASET STATISTICS.

Dataset	Yelp			Op Spam	
	CHI	NYC	ZIP	Positive	Negative
#Users	38,063	160,225	260,277	-	-
#Products	201	923	5044	20	20
#Spam Reviews	8,919	36,885	80,466	400	400
#Non-spam Reviews	58,477	322,167	528,133	400	400
%Spam	13.23%	10.27%	13.22%	-	-

binary labels: N representing genuine reviews and Y representing spam reviews.

2) **Op_spam_v1.4** from [28] (Table II), consists of truthful and deceptive hotel reviews of 20 Chicago hotels. The label of each review in Op_spam_v1.4 was gathered from Amazon's popular Mechanical Turk crowdsourcing service and five popular online review communities: Expedia, Hotels.com, Orbitz, Priceline, and TripAdvisor. Note that reviewer features are not available for the Op_spam_v1.4 dataset.

B. Baseline Models and Settings

We compare our proposed method, GDFN, with strong state-of-the-art baseline methods, including feature-centric and some recently proposed network-based models for spam review detection. The comparative models are:

- **NB** [5]: A naive Bayes classifier [34] based on four groups of features: content features, sentiment features, product features and meta data features
- **SVM+Ngram+BF** [4]: A standard n-gram (n=3) text categorization technique applied to detect negative deceptive opinion spam with SVM classifier
- **SpEagle** [26]: A pair-wise Markov Random Field model defined to tackle spam review detection task that utilized clues from metadata as well as relation data
- **CNN** [27]: A CNN method adopted to learn the textual information, and capture complex global semantic features for detecting spam reviews
- **CATS** [7]: A Xgboost [35] model as the classifier in the detector with multiple cross-platform independent features
- **HFAN** [6]: A Hierarchical Fusion Attention Network (HFAN) to automatically learn the semantics of reviews from user and product attribute
- **GAS** [18]: An end-to-end GCN-based Anti Spam (GAS) algorithm which incorporates the local context and the global context of comments with TextCNN classifier [36] to detect spam advertisements

We also use the pre-trained BERT-base model to exploit the information encoded in these pre-trained language models. We name this methods asGDFN (+BERT). To this end, we use the BERT-base multilingual cased pre-trained BERT model [1], which contains 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters. Since most of the review text contains multiple sentences, we use BERT-as-service [2] as a sentence encoding service, i.e., mapping a variable-length sentence to a fixed-length vector.

To compare our method with the traditional review mining methods, we have used commonly used evaluation metrics for this task, such as Average Precision (AP), Area Under Curve (AUC), Precision (Prec.), Recall (Rec.), and F1 measure (F_1). Specifically, for Yelp full datasets, AP and AUC are used as evaluation metrics. For Op_spam_v1.4 datasets, we evaluate Prec., Rec., and F_1 scores over two categories: negative and positive, respectively. For a fair comparison, we apply datasets with abundant metadata and profiles including conducting a five-fold cross-validation.

a) Data Preparation: Most of our pre-processing strategy has been widely used in the literature [6], [26]. The maximum length of reviews in Yelp full datasets is set to 200, and for Op_spam_v1.4 dataset, the maximum length is set to 100. We also compute some additional features which usually have been shown to improve performance. These are listed as below:

- Word Count of the documents – total number of words in the documents
- Character Count of the documents – total number of characters in the documents

[1]https://storage.googleapis.com/BERT_models

[2]https://github.com/hanxiao/BERT-as-service#1-download-a-pre-trained-BERT-model

TABLE III
SPAM DETECTION RESULTS ON WHOLE YELP AND OPSPAM DATASETS IN %. (BOLD INDICATES IMPROVEMENT OVER 10%)

Method	YelpCHI		YelpNYC		YelpZIP		Positive OpSpam			Negative OpSpam		
	AP	AUC	AP	AUC	AP	AUC	Prec.	Rec.	F1	Prec.	Rec.	F1
SVM+Ngram+BF	36.12	69.97	51.47	71.76	52.11	64.87	56.68	68.01	61.83	75.18	58.72	65.94
SpEagle	32.36	78.87	24.60	76.95	33.19	79.42	71.41	53.61	52.18	64.53	75.77	57.40
CATS	58.51	74.43	59.37	75.72	53.53	73.77	62.46	78.51	69.57	60.50	83.17	70.05
NB+Ngram	70.89	71.41	67.88	60.90	66.81	61.11	72.57	76.17	74.33	76.91	75.95	76.42
CNN	65.32	75.91	63.34	76.18	62.25	76.67	73.73	78.80	67.54	62.12	75.13	65.72
HFAN	48.87	83.24	53.82	84.78	62.35	87.28	86.96	67.31	75.88	61.17	40.00	48.37
GAS	68.90	71.02	70.09	71.67	67.02	60.00	88.65	84.61	81.53	88.60	84.87	81.63
GDFN	81.35	85.35	81.78	86.42	80.24	87.67	88.67	90.45	90.28	88.72	93.78	90.18
GDFN (+BERT)	**82.39**	87.69	**82.46**	87.85	**82.91**	88.05	88.75	92.83	**90.75**	89.82	**94.90**	**91.62**
Improvement(%)	16.22	5.34	17.65	3.62	23.70	0.88	0.11	9.72	11.30	1.38	11.82	12.24

- Average Word Density of the documents – average length of the words used in the documents
- Puncutation Count in the Complete Essay – total number of punctuation marks in the documents
- Upper Case Count in the Complete Essay – total number of upper count words in the documents
- Title Word Count in the Complete Essay – total number of proper case (title) words in the documents
- Frequency distribution of Part of Speech Tags: Noun Count, Verb Count, Adjective Count, Adverb Count and Pronoun Count.

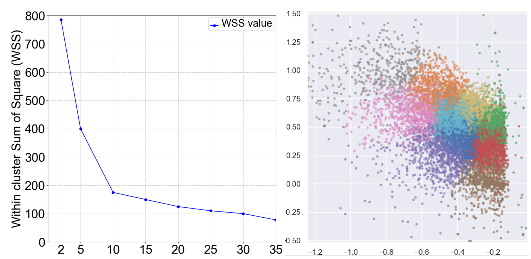
These features are applied as source input to the model during the training process.

b) Model Training: In the feature-based baselines, we make use of text and label. Review text is transformed into feature vectors. Each word is first represented by a 300-dimensional GloVe[3] [37] embedding of the word. For the CNN-based model, we configure 200 hidden layers and "mean" aggregation operation. Moreover, the rate of dropout is 0.25, and the training iterations are set to 200 epochs, with early stopping when the validation loss stops decreasing by 20 epochs. In the training process of the GCN-based method, the dropout rate is 0.5, L_2 loss regression is $2.5e-4$. In our model training, we adopt unsupervised learning for the clustering module and convolutional operation for the GCN-based module.

For the clustering module, we select the top 10 clusters of the unsupervised learning of review text, to make enough nodes for each review cluster (here we set up the minimum node number of each cluster as 200). We provide a visualization for the distribution of the review clustering graph in the embedding space where the figure illustrates the embedding space learned by the spectral clustering method. In Figure 2, we represent the top 10 clusters by unsupervised learning from the review source. Moreover, we utilize k-means as our clustering method and compute the symmetric normalized Laplacian.

This visualization is conducted to prove the similarity of the review source. As shown in Figure 2b, for unimodal interactions, obviously review text modality is the most predictive for the majority of samples, which is reasonable since the content is the most important clue for spam review analysis.

[3]https://nlp.stanford.edu/projects/glove/



(a) (b)

Fig. 2. **(a):** Compactness measure via WSS. **(b):** Visualization for distributions of 10 clusters in learnt embedding space. The different colors dots represent different review text clusters.

Furthermore, we have found that a small defined number of clusters may increase the computational complexity of graph construction and then lead to a lower clustering precision, while a large number of clusters do not show the difference among clusters reliably, the cluster boundaries are not very distinctive.

We utilize **W**ithin cluster **S**um of **S**quare (WSS) [38] technique as our metric for deciding the number of clusters. As shown in Figure 2a, we notice that our WSS measure drops considerably when the number of clusters is increased from 2 to 5, and again from 5 to 10, but the performance drop is comparatively lesser. Once we reach 10 clusters, the algorithm generally finds reliable groupings.

C. Results

Our experimental results are reported in Table III. We can see that the graph-based methods outperform feature engineering methods since the graph-based methods better capture intricate representations of spam reviews. They are also suitable to capture generalized features and interaction among multiple modalities.

Our method outperforms GraphSAGE [14] model, GAS [18], which justifies the advantage of combining graph structure and hybrid fusion strategy for spam detection. Additionally, CNN-based method cannot capture data with the graph structure, whereas HFAN, the hierarchical fusion network ignores important propagation features for unseen data prediction. This shows that obtaining graph structure information and fusion strategy separately, results in lower performance on spam detection.

TABLE IV
RESULTS OF ABLATION STUDY OF GDFN ON SPAM DETECTION PERFORMANCE (AVERAGE PRECISION IN %).

GDFN variants	YelpCHI	YelpNYC	YelpZIP
GDFN$_{ur}$	75.32	73.25	76.42
GDFN$_{ir}$	70.49	71.59	74.50
GDFN$_{ro}$	69.01	67.18	69.55
GDFN(+BERT)$_{ur}$	78.25	78.28	79.02
GDFN(+BERT)$_{ir}$	74.21	73.80	78.42
GDFN(+BERT)$_{ro}$	73.48	68.55	73.21

CNN only uses the convolutional hidden layer to capture feature vectors from Euclidean structure data so it is dependent on data samples. However, the review platform is similar to a social network. Unlike the CNN, GCNs enable the proposed model to pay more attention to the non-Euclidean structural information of the review posts, which helps improve our model's performance. Further, the experiment result of the GCN-based model, GAS, has shown a significant fluctuation on different datasets, which makes it less ideal and overfits in case of some input samples, e.g., GAS obtains a better performance on OpSpam positive datasets. The proposed fusion strategy fuses extra information from user-item level to influence the final prediction, which helps us get a relatively stable result.

D. Ablation Study

To analyze the effect of the individual components of GDFN, we conduct an ablation study where we consider three different components: GDFN$_{ur}$ which includes user-review text only, GDFN$_{ir}$ which includes item-review only, and GDFN$_{ro}$ which includes review text without user and item information.

As is shown in Table IV, we have observed that GDFN$_{ur}$, GDFN$_{ir}$ and GDFN$_{ro}$ cannot outperform our main model (results in Table III. Meanwhile, GDFN$_{ur}$'s performance is close to that of GDFN, demonstrating that user-level information plays an important role in spam detection. We also observe that the worst results obtained from the variant GDFN$_{ro}$, but these results are still better than most of the other baseline methods, showing the superiority of our proposed framework for spam review detection.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel model named GDFN, to predict spam reviews based on a unimodal graph to cluster similar review text for extracting aggregation-based semantic features and then encode user (item)-level information to strengthen the final representation. We also utilize the fusion mechanism to obtain the inherent relationship among users, reviews, and items. To evaluate the performance of GDFN, we conducted a series of experiments on two public datasets, to demonstrate the superiority of the model in comparison with state-of-the-art models.

Given the recent success of multimedia sharing platforms, the items posted on these online social media websites contain rich multimedia information (e.g., visual and acoustic). Exploiting these multi-modality features is an interesting future direction. Moreover, data connections can be more complex than a pairwise relationships on the social networks. Addressing this problem in hypergraph networks can be considered a new research line in this field.

ACKNOWLEDGMENTS

This work is partially supported by the Australian Research Council (ARC) under Grant numbers DP22010371, LE220100078, DP200101374, and LP170100891. Shoaib Jameel is supported by NVIDIA Academic Hardware Grant.

REFERENCES

[1] N. Jindal and B. Liu, "Opinion spam and analysis," in *WSDM*, 2008, pp. 219–230.

[2] J. Li, X. Wang, L. Yang, P. Zhang, and D. Yang, "Identifying ground truth in opinion spam: an empirical survey based on review psychology," *Applied Intelligence*, pp. 1–16, 2020.

[3] Y. Wu, E. W. Ngai, P. Wu, and C. Wu, "Fake online reviews: Literature review, synthesis, and directions for future research," *Decision Support Systems*, p. 113280, 2020.

[4] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 497–501. [Online]. Available: https://www.aclweb.org/anthology/N13-1053

[5] F. H. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *AAAI*, 2011.

[6] C. Yuan, W. Zhou, Q. Ma, S. Lv, J. Han, and S. Hu, "Learning review representations from user and product level information for spam detection," *ICDM*, vol. 2019-Novem, pp. 1444–1449, 2019.

[7] H. Weng, S. Ji, F. Duan, Z. Li, J. Chen, Q. He, and T. Wang, "Cats: Cross-platform e-commerce fraud detection," in *ICDE*, 2019, pp. 1874–1885.

[8] C.-C. Wang, M.-Y. Day, C.-C. Chen, and J.-W. Liou, "Detecting spamming reviews using long short-term memory recurrent neural network framework," in *E-commerce, E-Business and E-Government*, 2018, pp. 16–20.

[9] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE TNN*, vol. 20, no. 1, pp. 61–80, 2008.

[10] G. Wang, S. Xie, B. Liu, and S. Y. Philip, "Review graph based online store review spammer detection," in *ICDM*, 2011, pp. 1242–1247.

[11] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "Netspam: A network-based spam detection framework for reviews in online social media," *Trans on IFS*, vol. 12, no. 7, pp. 1585–1595, 2017.

[12] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *CIKM*, 2018, pp. 2077–2085.

[13] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv*, 2018.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017, pp. 1024–1034.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.

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

[17] S. Ali Alhosseini, R. Bin Tareaf, P. Najafi, and C. Meinel, "Detect me if you can: Spam bot detection using inductive representation learning," in *WWW*, 2019, pp. 148–153.

[18] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *CIKM*, 2019, pp. 2703–2711.

[19] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD*, 2018, pp. 974–983.

[20] L. He, H. Chen, D. Wang, S. Jameel, P. Yu, and G. Xu, *Click-Through Rate Prediction with Multi-Modal Hypergraphs*. New York, NY, USA: Association for Computing Machinery, 2021, p. 690–699. [Online]. Available: https://doi.org/10.1145/3459637.3482327

[21] L. He, D. Wang, H. Wang, H. Chen, and G. Xu, *TagPick: A System for Bridging Micro-Video Hashtags and E-Commerce Categories*. New York, NY, USA: Association for Computing Machinery, 2021, p. 4721–4724. [Online]. Available: https://doi.org/10.1145/3459637.3481979

[22] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, "Graph convolutional networks with markov random field reasoning for social spammer detection," in *AAAI*, vol. 34, no. 01, 2020, pp. 1054–1061.

[23] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *AAAI*, vol. 34, no. 01, 2020, pp. 549–556.

[24] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering." in *AAAI*, 2016, pp. 1969–1976.

[25] T. Baltrusaitis, C. Ahuja, and L. P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," *TPAMI*, vol. 41, no. 2, pp. 423–443, 2019.

[26] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *KDD*, 2015, pp. 985–994.

[27] X. Wang, K. Liu, and J. Zhao, "Handling cold-start problem in review spam detection by jointly embedding texts and behaviors," in *ACL*, 2017, pp. 366–376.

[28] Y. R. Chen and H. H. Chen, "Opinion spammer detection in web forum," 2015, pp. 759–762.

[29] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *WWW*, 2011, pp. 675–684.

[30] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[31] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *TPAMI*, vol. 41, no. 2, pp. 423–443, 2018.

[32] S. Rendle, "Factorization machines," in *ICDM*, 2010.

[33] S. Mai, H. Hu, and S. Xing, "Modality to modality translation: An adversarial representation learning and graph fusion network for multimodal fusion," in *AAAI*, vol. 34, no. 01, 2020, pp. 164–172.

[34] I. Rish *et al.*, "An empirical study of the naive bayes classifier," vol. 3, no. 22, pp. 41–46, 2001.

[35] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*, 2016, pp. 785–794.

[36] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv*, 2014.

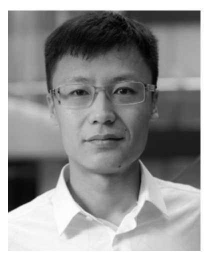
[37] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[38] T. Dao, K. Duong, and C. Vrain, "A filtering algorithm for constrained clustering with within-cluster sum of dissimilarities criterion," in *ICTAI*, 2013, pp. 1060–1067.

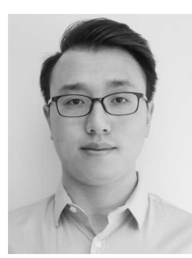
Li He Li He received the Master degree in computer science from the University of Science and Technology of China, China. He is currently a Ph.D candidate in the School of Computer Science, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include network science, graph theory, information network representation learning, user behavior modelling, and social network mining.

Guangdong Xu Guandong Xu received the Ph.D. degree in computer science from Victoria University, Footscray, VIC, Australia, in 2009.,He is currently a Full Professor with the School of Computer Science, University of Technology Sydney, Ultimo, NSW, Australia. He has published 3 monographs in Springer and CRC press and more than 250 journal articles and conference papers in data science, recommender systems, text mining, and social network analysis. Dr. Xu has served as a Guest Editor for Pattern Recognition, the IEEE Transactions on Computational Social Systems, Journal of Software and Systems, World Wide Web journal, Multimedia Tools and Applications, and Online Information Review. He is also an Assistant Editor-in-Chief of World Wide Web journal.

Shoaib Jameel Shoaib Jameel received the Ph.D. degree from The Chinese University of Hong Kong. He is currently a Lecturer in computer science and artificial intelligence at the School of Computer Science and Electronic Engineering, University of Essex, U.K. He works with various technology startups in the U.K., spearheading their technical sphere, where his research outputs are directly applied to their production systems. His works have appeared in various prestigious conferences and journals, such as SIGIR, AAAI, ACL, IJCAI, and TOIS. His research interests include text mining, natural language processing, and computer vision. He is a fellow of the Higher Education Academy



Xianzhi Wang Xianzhi Wang is currently a lecturer with the School of Computer Science, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include Internet of Things, data fusion, and machine learning. He received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2014. He is a Member of the IEEE and the ACM. Contact him at xianzhi.wang@uts.edu.au.



Hongxu Chen Hongxu Chen received the PhD degree in computer science from the University of Queensland, Australia. His research interests include network science, graph theory, information network representation learning, recommender systems, social user modelling, and social network mining.