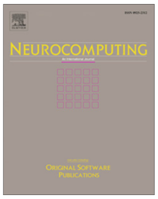
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Predicting video engagement using heterogeneous DeepWalk

Iti Chaturvedi [a],*, Kishor Thapa [a], Sandro Cavallari [b], Erik Cambria [b], Roy E. Welsch [c]

[a] College of Science and Engineering, James Cook University, Australia
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[c] Sloan School of Management, MIT, United States

ARTICLE INFO

Article history:
Received 23 February 2021
Revised 2 July 2021
Accepted 29 August 2021
Available online 8 September 2021
Communicated by Zidong Wang

Keywords:
Video engagement
DeepWalk
Online behavior
One-class model

ABSTRACT

Video engagement is important in online advertisements where there is no physical interaction with the consumer. Engagement can be directly measured as the number of seconds after which a consumer skips an advertisement. In this paper, we propose a model to predict video engagement of an advertisement using only a few samples. This allows for early identification of poor quality videos. This can also help identify advertisement frauds where a robot runs fake videos behind the name of well-known brands. We leverage on the fact that videos with high engagement have similar viewing patterns over time. Hence, we can create a similarity network of videos and use a graph-embedding model called DeepWalk to cluster videos into significant communities. The learned embedding is able to identify viewing patterns of fraud and popular videos. In order to assess the impact of a video, we also consider how the view counts increase or decrease over time. This results in a heterogeneous graph where an edge indicates similar video engagement or history of view counts between two videos. Since it is difficult to find labelled samples for 'fraud' video, we leverage on a one-class model that can determine 'fraud' videos with outlier or abnormal behavior. The proposed model outperforms baselines in F-measure by over 20%.

1. Introduction

Video marketing and analysis has become increasingly popular in recent years thanks to 5G but also thanks to advancements in multimodal processing [9,31,26,8]. In this paper, we aim to study the online behavior of consumers on YouTube, where videos are hosted and shared among friends and subscribers. Several brands display their advertisement in the beginning of a YouTube video. The advertiser pays money proportional to the number of views and likes that his advertisement receives. However, recently there has been an explosion of fake views that are generated by a robot. It is very difficult to label such fraudulent activity using a human expert. Hence, in this paper we consider a heterogeneous engagement model that can combine different types of metrics such as viewing duration and view counts. Since, less than 1% of videos are targets of fraud this results in an imbalanced dataset. We can use a one-class model to guide the learning of latent embeddings or communities in a network of videos.

Fig. 1 shows an example of a YouTube Advertisement. The view count is shown as over 500K and the number of likes is 3.3K. In order to understand the engagement, we also need to look at the percentage of viewing duration using a software that can record actions such as play, pause and rewind. Fake views will result in spikes in the view counts and will disappear over time. Hence, in this paper if the total view count for a video is above a threshold after three months, we label it as a popular video [21]. The collection of online behavior is dependent on the availability of a strong internet connection. The one-class model predicts contours that minimize the distance of all videos from the origin. View counts that lie outside a contour can be discarded as noisy or fake views [32,22].

Our problem is similar to advertisement click fraud where a robot keeps clicking on products and does not make a purchase [14]. In [28], the authors used a binary tree where leaf nodes are IP address to track clusters of malicious websites. However, they concluded that IP address is a poor indicator as bots may keep getting discovered and removed. Instead, we rely on the viewing patterns of each video over time. DeepWalk is a popular algorithm for community detection in social networks [24,6]. For a given input network it can learn communities that maximize the probability of a random path through the network [5]. Heterogeneous DeepWalk is challenging due to the presence of nodes of different types [11]. Here, in order to deal with the imbalanced community problem, we employ a one-class model to guide the selection of random paths through the network. The resulting model is referred to as Heterogeneous Engagement Auxiliary DeepWalk (HEAD).

* Corresponding author.

https://doi.org/10.1016/j.neucom.2021.08.127
0925-2312/© 2021 Elsevier B.V. All rights reserved.

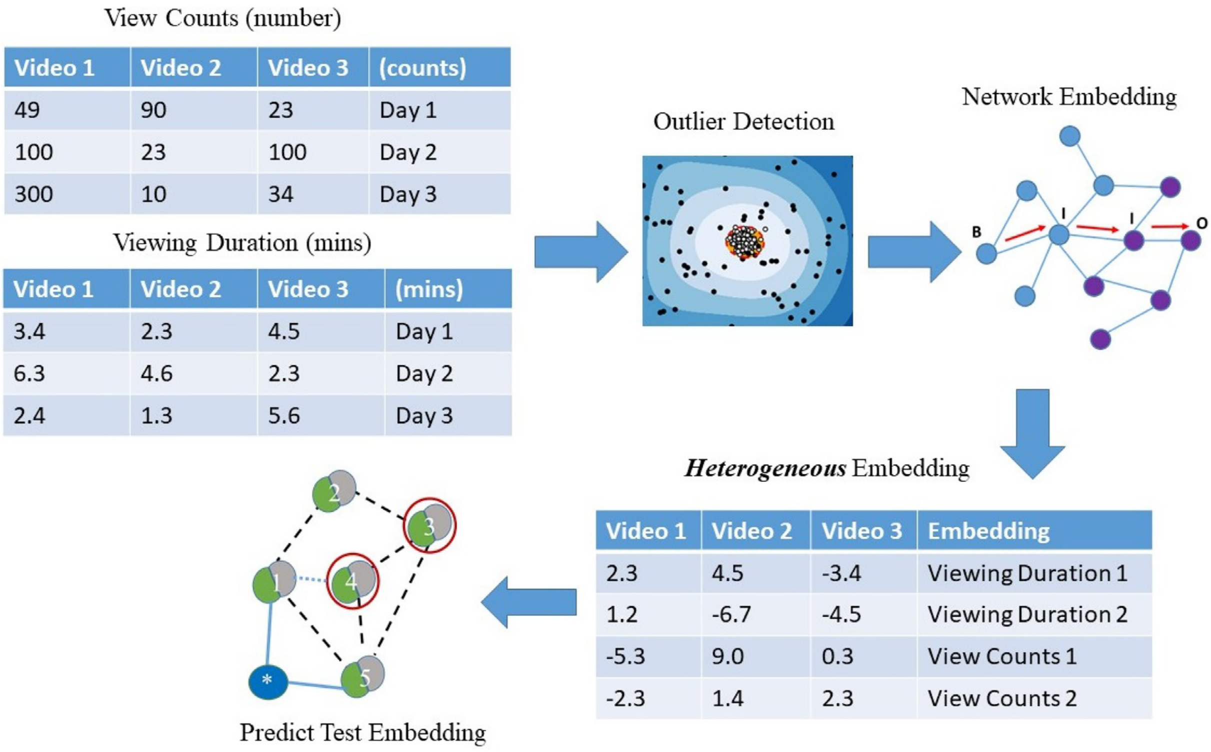
Fig. 1. Example of a YouTube Advertisement. The number of views, likes and dislikes are recorded over time. For capturing video engagement, a software to record actions such as play, pause or rewind is used.

The Fig. 2 illustrates the flowchart for the proposed HEAD algorithm. We consider two different types of dataset for each video namely the view counts and the viewing duration on each day. Each video is labelled as ‘fraud’ or ‘popular’ based on the total view counts at the end of three months. Next, a one-class model is used to identify outlier videos that will hinder the convergence of the multimodal DeepWalk model. Next, we construct a network of videos where edges are allowed between videos with the same labels and also high covariance. A few edges are allowed between borderline videos. DeepWalk determines the embedding corresponding to the highest probability path through the network that also corresponds to the global maxima of the convex polytope for the sequence of videos. We concatenate the embeddings for both types of dataset into a single vector. Lastly, we can use this heterogeneous embedding to predict the embedding and label for any new video for which either the viewing durations or view counts are known.

The organization of the paper is as follows: Section 2 reviews related works and datasets on video engagement; Section 3 provides the preliminary concepts necessary to understand the present work such as the DeepWalk and One-class model; Section 4 introduces the proposed heterogeneous model for the fusion of video engagement and view count data; Section 5 validates the proposed method on two real-world datasets; finally, Section 6 provides concluding remarks.

2. Related work and contributions

In recent years, video marketing and multimodal analysis have raised growing interest within both the scientific community, for the many exciting open challenges, as well as the business world, due to the remarkable benefits to be had from marketing and financial prediction [15,4,7,34]. Unfortunately, this has also led to the emergence of ‘fraud’ videos. Marciel et al. [19] showed that portals


View Counts (number)

Video 1	Video 2	Video 3	(counts)
49	90	23	Day 1
100	23	100	Day 2
300	10	34	Day 3

Viewing Duration (mins)

Video 1	Video 2	Video 3	(mins)
3.4	2.3	4.5	Day 1
6.3	4.6	2.3	Day 2
2.4	1.3	5.6	Day 3

Heterogeneous Embedding

Video 1	Video 2	Video 3	Embedding
2.3	4.5	-3.4	Viewing Duration 1
1.2	-6.7	-4.5	Viewing Duration 2
-5.3	9.0	0.3	View Counts 1
-2.3	1.4	2.3	View Counts 2

Fig. 2. This figure illustrates the flowchart for the proposed HEAD algorithm. Starting with two different types of dataset namely viewing durations and view counts, we can predict the label for any new video. First, we remove outliers using a one-class model. For the remaining videos latent embeddings are determined using DeepWalk. The embeddings for different data types are concatenated resulting in a heterogeneous representation for each video. Lastly, we can use network regression to label a new video as fraud or popular.

such as YouTube are susceptible to attacks. They used a probe to generate views for a video from different IP addresses and then compared it to the number of view counts. The pattern of view generated by a software bot will be different from that of a real human being. Hence, we propose a multimodal classifier for predicting fraudulent video advertisements. Advertisements with high engagement can also be semantically connected to the content of a YouTube video [33].

DeepWalk is an algorithm to learn structural regularities using short random walks in a given network [23]. Each node in the network is represented by a latent feature vector that captures the neighborhood similarity and community membership. Such a model is unable to accommodate new users and assumes all the edges have the same similarity metric. In this paper, we overcome these challenges using network regression to predict features for any new user or modality.

Most previous graph embedding methods focus on homogeneous networks consisting of a single type of nodes and links [17,27]. In [25], the authors proposed fusion methods for combining different types of data in a heterogeneous network. They propose the use of semantic meta paths to constrain the embedding. However, such paths are often not known. Hence, in this paper, we introduce a small number of random edges between different types of data nodes. For the remaining video nodes, we only allow edges between videos with similar view counts and online behavior. Our work is inspired by the deep clustering algorithm proposed in [30]. Here, a reconstruction loss for predicting graph structure using an attention auto-encoder is used during clustering. In contrast, we first cluster the data using a one-class model to remove outliers, next we predict the embedding for the best clustering.

In [29] the authors recommend friends in a heterogeneous social network where an edge can represent friendship, contact or chat relationships. They use DeepWalk on each homogeneous sub-network and then combine the learned embeddings using a neural network. Such a method can only predict two existing users. In contrast, we use covariance between embeddings to determine the heterogeneous neighbors for a new consumer and network regression to predict the embedding.

Predicting clicks accurately has widespread application in advertising and real-time bidding. It is necessary to make predictions billions of times per day and update the model as new clicks and non-clicks are observed. In [14] the authors aim to predict if a particular link is clicked or not. They use an online probit regression model where the probability that a link is clicked is sampled from a Gaussian distribution. The mean and variance of the distribution are updated with each new sample. To ensure stability they consider pruning of weights. Instead, in [20] the authors used regularization to ensure sparsity in an online logistic loss model. An explosion of information might make it difficult for users to select the right content due to information overloading. Here, it is useful to prune the content and select only the best web pages. We can also include information such as location and user fatigue while predicting clicks. In [1] the authors show that there is an increased propensity to click the same link.

In order to improve the accuracy of heterogeneous auxiliary networks, a better fusion of extracted features was proposed in [16,10]. They also show that it is better to simultaneously tackle multiple tasks such as optical flow estimation and lane detection during autonomous driving. Similar to their approach we consider the simultaneous extraction of embeddings from both viewing duration and view counts.

The following is a summary of the significance and contributions of the research work presented in this paper:

- Only a fraction of videos can be labelled as fraud resulting in an imbalanced dataset. We propose to use a one-class model that can be trained on only samples from a single class. Videos that lie far away from the origin are outliers that can be discarded prior to training.
- In order to model the online behaviour of consumers we consider a time series of view counts and total viewing duration over three months. We can leverage on the sequence of videos viewed by a consumer in a particular channel to approximate the heterogenous convex objective function.
- We learn the embedding separately for both view counts and viewing durations. These are concatenated resulting in a heterogeneous embedding. For a new test video, we can use covariance to determine the closest neighbours in the training set. Next, network regression can be used to predict the embedding and class label without the need for retraining.
- We show that the heterogenous latent embeddings learned by the model are easily clustered into two communities where one of them can be labelled as fraud videos using training labels. Fraud videos are fake viewing patterns generated by a bot.

Validation of the method is performed on two benchmark datasets. Due to lack of available benchmark datasets for video engagement, we validate the model on a benchmark dataset for click fraud detection. A fraud person will randomly browse for products without any purchases. In contrast, a genuine customer will actively compare products of a certain type and make a purchase. This dataset is from a ‘Talking Data’ challenge that aims to identify a fraud from a genuine customer who will buy an app. Here, we consider the sequence of clicks or apps viewed from a specific IP address. Each click is represented by the app id that was viewed. The click sequence is labelled as ‘fraud’ or ‘buy’ depending on whether there is a purchase or not. Next, we consider a heterogeneous dataset of viewing durations and view count history for YouTube advertisements collected at a university campus. The view counts represent the evolution of popularity over time and the video sequence corresponds to duration in minutes spent by a user on a video.

The click fraud dataset has samples of fraud robots that randomly browse apps on the website without actually purchasing anything. Genuine customers are labelled based on their account profiles and purchases. There is however no visible difference in the pattern of clicks. Similarly, the only way to annotate a real view as opposed to a fake one is to verify the credibility of the account owner. The view counts of a fraud advertisement will have spikes that are generated by a bot and the corresponding engagement in the form of viewing durations will be low. It could also take the form of pixel stuffing where the advertisement is never visible to users. Such spikes will however not determine the long term influence of the video. Hence, in this paper we used the total view counts at the end of three months to categorize a video as popular or fraud.

3. Preliminaries

In this section, we introduced some theoretical models for classifying videos in a social network based on the level of engagement. The aim of the model is to differentiate fraudulent online behavior that is generated using a bot from that of a human. We consider a temporal regression model to label each video as ‘fraud’ or ‘popular’. We show that the error between the true and predicted label in the training videos can be used to learn the influence of each neighbor in the social network [3]. In order to efficiently learn labels of a large number of videos, we use a heuristic approach called DeepWalk [28]. Here, each video is represented by a vector of features such that videos in the same community lie close to each other in the vector space.

3.1. Online behavior

Online user behavior can be recorded in the form of view counts or a video viewing durations. Let us consider a sequence of advertisements that appear on a particular channel given by $\mathrm{x} = (x_1, x_2, \ldots, x_n)$ where x_i is a vector of observations and n is the number of videos in that channel. Each video is classified as 'fraud' or 'popular'. Hence, the class label $y_i \in \{0, 1\}$. We can use the label of the previous advertisement in a channel to predict the label of the current advertisement. The resulting model can be defined as follows:

$$\begin{aligned} h_i &= f(h_i) + g(h_i)h_{i-1} \forall i = 1, 2, \ldots, \mathrm{n} - 1 \\ h_1 &= f(h_1) + g(h_1)x_i + \mathrm{b} \\ y_i &= h_n \end{aligned} \tag{1}$$

where h_i is the latent representation of x_i and b is the bias that is learned using gradient descent. The functions $f(h)$ and $g(h)$ define the activation function for input neurons and the inter-connected neurons respectively. Hence, $f(h_i)$ is a function mapping each video to the output label y_i and $g_(h_i)$ is a function mapping each video h_i given the previous video h_{i-1} in the sequence to the channel label y_i. Here, the index i corresponds to the position of a video in the sequence and ranges from 1 to n. The latent representation h_i for each video is dependent on the previous video h_{i-1}. Hence, videos in the same community will influence the label of each other. To determine this latent embedding h_i we propose to use DeepWalk in the next section.

We consider a sequence of videos in the social network such that each pair in the sequence is connected. The label y_i for video i in the sequence is dependent on the weighted sum of the previous n videos. The latent representation of each video i is denoted by the vector h_i. Hence, h_n is the latent feature vector for the penultimate video in the sequence.

In order to learn the parameters $\theta = (\mathrm{h}, b)$, we update using the ground truth y^*:

$$\mathrm{e} = \mathrm{y}^* - \mathrm{y}\Delta\theta = \frac{\partial \mathrm{e}}{\partial \theta} = \frac{\partial \mathrm{e}}{\partial f(\mathrm{h})} \cdot \frac{\partial f(\mathrm{h})}{\partial \theta} \tag{2}$$

where $f(h)$ is the mapping function to latent space.

3.2. DeepWalk

In this paper, we consider the use of DeepWalk to determine the mapping function $f(h)$ for videos in a given channel. As the name suggests DeepWalk finds the sequence of videos (also known as a walk) that has highest posterior probability given a particular class label. This means that in DeepWalk 'fraud' videos would lie on the same path or sequence as defined above using Eq. (1).

DeepWalk is a method that learns a latent space representation of social interactions in a graph of users [23]. For example, in Fig. 3 we illustrate a network with two communities denoted by different colors. The representation learned by DeepWalk aims to encode the community structure into a vector space such that communities are well separated. We can note the correspondence between the community structure in the input graph and the embedding learned.

Consider a graph G with a set of nodes X that are connected by a set of edges. The aim of DeepWalk is to classify the nodes of the graph into one or more categories. Unlike traditional machine learning here we utilize the dependence of the examples embedded in the structure of G to label the graph. Previous authors considered approximate inference algorithms such as Gibbs Sampling and label relaxation to compute the posterior distribution of labels given the network structure [13]. Instead, DeepWalk is an unsupervised method that learns features that capture the graph structure independent of the label distribution.

Our goal is to determine a small number of latent dimensions $m \times d$ from the complete graph adjacency matrix of dimension $m \times m$. These d low-dimensional representations are distributed meaning that each social phenomenon is expressed by a subset of dimensions. We denote a random walk rooted at vertex h_i as a stochastic process with random variables $h_{i+1}, h_{i+2}, \ldots, h_{i+\mathrm{n}}$ such that h_{i+1} is a vertex chosen at random such that an edge exists with h_i and n is the length of the random walk. We can rephrase the problem of estimating the class of a vertex given all previous vertices in a random walk as a language model.

In a language model we consider a sequence of words $(h_i, h_{i+1} \ldots, h_{i+n})$ appearing in a corpus and the aim is to maximize $p(h_i|h_{i+1}, \ldots, h_{i+n})$ over the whole training corpus. However, our goal is to learn a latent representation, not only a probability distribution of node co-occurrences, so we introduce a mapping function $f(h_i) \in \mathrm{R}^{\mathrm{m} \times d}$ where R is the set of real numbers. This mapping f represents the latent social representations associated with each vertex h_i in the graph. Furthermore, the context is composed of words appearing to the left and the right side of a word.

Here we consider a heterogeneous model that can combine different types of datasets. The time-series dataset can be approximated using a Taylor series of a convex polytope. Here, the gradient of each video in a sequence is dependent on the previous n videos. The resulting gradient is piecewise linear and is able to model complex multimodal datasets. DeepWalk uses maximum posterior probability to determine the sequence of videos in the convex polytope corresponding to a global maxima. It is previously shown that for a convex polytope of n vertices, the Taylor series approximation is a summation over piecewise gradients of the vertices. We can hence rewrite Eq. (2) as follows:

$$\begin{aligned} \Delta\theta &= \sum_{k=i-n/2}^{i+n/2} \frac{\partial \mathrm{e}}{\partial f(\mathrm{h}_k)} \cdot \frac{\partial f(\mathrm{h}_k)}{\partial \theta} \\ \theta &= \arg\max_\theta - \log p(\mathrm{h}_{i-\mathrm{n}/2}, \ldots, \mathrm{h}_{i-1}, \mathrm{h}_{i+1}, \ldots, \mathrm{h}_{i+\mathrm{n}/2} | f(h_i|\theta)) \end{aligned} \tag{3}$$

where the maximum probability path of h vertices in the graph corresponds to the global solution of θ. It is reasonable to assume that such a piecewise solution is able to handle multimodal functions in heterogenous datasets well.

Computing the partition function for this equation is expensive. We can assign the vertices to the leaves of a binary tree, then the prediction problem turns into maximizing the probability of a specific path in the tree. We can speed up the training process by assigning shorter paths to the frequent vertices in the random walks. In this paper, we use DeepWalk to predict the interactions between users in a classroom or consumers of a product. We can use the time series of click sequences collected online to determine the similarity or edges between users and then DeepWalk is used to identify communities of users with different behaviors or product preferences.

For a given network structure an edge between two nodes indicates they are from the same community and have high similarity. In order to reduce the dimensionality of the network, we represent each node as a vector such that similar nodes lie close to each other in the new vector space. In order to discover the neighborhood of a node, we consider a fixed number of random walks of length l starting at each node.

Each random walk is a sequence of nodes where the first node h_i is denoted by B (Begin) and the last node h_{i+n} is denoted by O

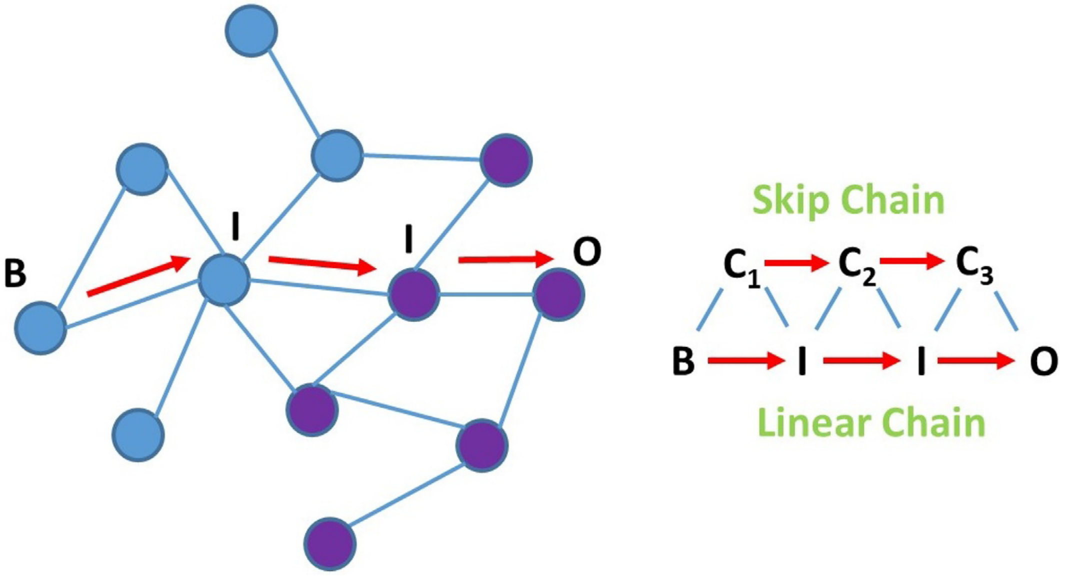


Fig. 3. Random walks in a network with two communities. The Skip-chain walk has a second-order dependency to the underlying Linear-chain.

(Outside). The nodes that lie on the path are denoted as I (Inside). Such a random walk is denoted as a linear chain of red arrows in Fig. 3. Next, we learn a low dimensional vector embedding such that the occurrence probability of each node given its immediate neighbor in the random walks is maximum using Eq. (3).

We might also be interested in the co-occurrence probability of a node given the previous two nodes in a random walk. This is achieved by introducing a constraint on the random walk path. In particular, each pair of consecutive nodes in the graph is represented by a new node $C_i = I_i, I_{i+1}$. Now, walks are only allowed from C_i to C_{i+1} such that $C_i = I_i, I_{i+1}$ and $C_{i+1} = I_{i+1}, I_{i+2}$. Next, we can maximize the co-occurrence probabilities of this new set of skip nodes C_i to compute the low dimensional vectors using Eq. (3).

Fig. 4 shows a sample network for the heterogeneous video data. We have two types of nodes (1) Fraud videos (2) Popular videos. Edges are only allowed among nodes of the same types if the covariance between them is above a threshold. Borderline videos can have edges to both fraud and popular videos.

4. Heterogeneous auxiliary DeepWalk

In this section, we show that network regression can be used to predict embeddings of new videos in a channel. In order to cope with the imbalanced samples available for fake videos, we also use a one-class model to remove outliers. Lastly, we describe the heterogeneous DeepWalk for combining embeddings learned from both video viewing and view count data.

4.1. Network regression

During testing we can predict the mapping function or embedding for a new video $x^{\dagger}$ using network regression. We can define the embedding of $x^{\dagger}$ as a weighted combination of its closest neighbors in the training set.

$$\begin{aligned} x^{\dagger} &= w_{\mathcal{N}} \tfrac{1}{\mathcal{N}} \sum_{j \in \mathcal{N}_i} x_j + b \\ x^{\dagger} &= \beta \mathrm{x}_{\mathcal{N}} + \mathrm{b} \end{aligned} \tag{4}$$

where $\mathcal{N}_i$ is the set of neighborhood nodes for video $x^{\dagger}, \beta$ are the coefficients for each neighbor and the noise term is b.

Using DeepWalk as described in the previous section we can compute β using the mapping function $f(\mathrm{x}_{\mathcal{N}})$.

4.2. One-class SVM

It is relatively easy to gather training data for popular videos. On the other hand, collection of samples for fraud videos is rare and sometimes impossible. Even if we simulate a fraud video, there is no way to guarantee that it will capture all the behavior of a robot. To cope with this problem, one class classification models have been developed. By just providing the popular videos as training data, it can create a representative model. During testing each sample that is too different from this model is labelled as out-of-class. In this paper, we consider a one-class support vector machine (SVM) model to represent the set of online users.

The traditional SVM defines a hyper-plane that partitions that training samples into two vector spaces corresponding to each class. The distance from the closest point from each class to the hyper-plane is equal, thus the constructed hyper-plane searches for the maximal margin between the classes. Furthermore, when the data is not linearly separable, we can project the data to a high dimensional space $f(\mathrm{x})$ where a hyper-plane exists. As an extension, the one-class SVM separates the data points from the origin and maximizes the distance from this hyper-plane $w^T\mathrm{x} = 0$ to the origin.

Hence, due to lack of labelled samples for fraud videos, we also consider a one-class model to determine the mapping function $f(x_i)$.

$$\begin{aligned} &\min_{w,\eta,\rho} \tfrac{1}{2} ||w||^2 + \tfrac{1}{v \times n} \sum_{i=1}^{n} \eta_i - \rho \\ &(w.f(x_i)) \geqslant \rho - \eta_i, \eta_i \geqslant 0 \qquad i = 1, 2, \ldots, n \end{aligned} \tag{5}$$

where $v \in (0, 1)$ controls the fraction of outliers and η is set of slack variables that can lie within the separation margin and avoid overfitting.

We can solve the above equation for w and variable ρ that gives the best accuracy on training data. A limitation of one-class SVM is that we can only determine the contours around origin in two dimensions. Hence, it is difficult to interpret the outliers in multiple dimensions. In this paper, we consider random pairs of users in

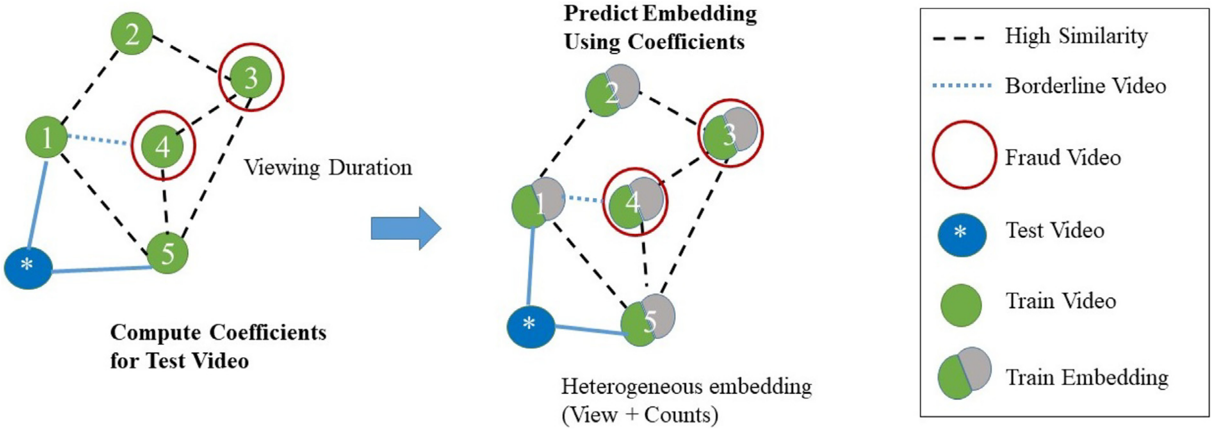


Fig. 4. This diagram illustrates the process of predicting embeddings for any new test video shown as a dark blue circle with a * notation. The first network is used to determine the neighborhood of the new test video using MSE. This can be used to predict the heterogeneous embedding (both video and view counts) of the new video in the second network. Edges are not allowed between a fraud and a popular video.

two dimensions and the one-class model that has the highest average accuracy for both classes. Here a contour is used to minimize the distance of samples from the origin. We can discard videos that are outliers in both 'viewing durations' and 'view counts' datasets. This will improve the convergence of DeepWalk in determining the convex polytope corresponding to a global maxima solution.

Algorithm 1. HEAD Algorithm

```
% Training of HEAD model
for For all videos $h_i$ do
    for For all videos $h_j$ do
        if covariance$(h_i, h_j) > \gamma$ then
            $G(i,j) = 1$
        else
            $G(i,j) = 0$
        end if
    end for
end for
DeepWalk: Embedding $f(\mathrm{h})_{m\times d}$
% Testing of HEAD model
$f(\mathrm{h}) = f(\mathrm{h}^v) \bigcup f(\mathrm{h}^c)$
for For all test videos $h_i^\dagger$ do
    for For all $h_j \in f(\mathrm{h})$ do
        if covariance$(h_i^\dagger, h_j) > \gamma$ then
            $G(i,j) = 1$
        else
            $G(i,j) = 0$
        end if
    end for
end for
Predict $f(h_i^\dagger)$ as weighted sum over $G$
c: $k$-Means clustering of $f(\mathrm{h})$
Classify $h_i$ to closest centroid in c
```

4.3. Heterogeneous framework

In order to run DeepWalk on the click sequence data, we need to create a network representation of the users. Here, we use time series Euclidean similarity to determine whether an edge exists between two users. In addition, we also use the known class label of the training data, edges are not allowed between videos from different class labels. Algorithm 1 details the HEAD algorithm. The algorithm takes as input the sequence of video viewing durations: h^v as well as the view count history: h^c for each video. The videos are labelled as 'fraud' or 'popular' based on the total view count. Each sequence for video i is denoted by a vector x_i and the label is $y_i \in \{0, 1\}$. The model has two steps (i) determine the outliers using one-class SVM (ii) determine the embedding using DeepWalk. For each contour predicting by one-class SVM we compute the overall accuracy of both classes on labelled training data. We select the contour with the highest accuracy as optimal for the dataset. Next, we eliminate these outliers from the dataset. Here, we assume that the online consumption of videos in the same channel or category is similar. We create a network G where each node is a video and the edges correspond to high similarity between two nodes. The similarity between the two nodes is determine using covariance of the collected dataset. We also use the total view count of a video during training to constrain the presence of an edge between them. For example, 'popular' videos may only connect to 'popular' peers. On the other hand, 'borderline' videos may connect to both 'popular' and 'fraud' videos. We convert the high-dimensional YouTube network to low-dimensional vector embeddings using DeepWalk. Now each video is represented by a new vector such that videos with similar view counts and consumption lie closer in the vector space.

During testing, we combine the graph embeddings from both types of dataset: $f(\mathrm{h}) = f(\mathrm{h}^v) \bigcup f(\mathrm{h}^c)$. Next, we create a heterogeneous similarity network G on the combined embedding using co-variance. For each test video $h^\dagger$ we can select the closest neighbor video in the original vector space. Next, we predict the embedding of the test video as a weighted sum over edges in G. The coefficients can be determined using the original dataset. Lastly, we cluster the learned embeddings of the training videos into 'fraud' and 'popular' using k-means. The test video is classified to the closest centroid among the two clusters. Here we used the prior knowledge that the two clusters should correspond to fake and popular videos. Hence, we set k to two. Furthermore, during initialization we randomly assigned one centroid to a popular video and another centroid to a fake video. We do not see any significant change in accuracy when different videos were chosen as starting centroids.

Fig. 4 illustrate the process of predicting embeddings for any new test video shown as a dark blue circle with a * notation. The first network is determined using training samples of viewing durations for each video. Videos with high covariance are connected by an edge. Videos with a red circle are annotated as 'fraud' based on the total view counts at the end of three months. Edges are only allowed among videos with the same label namely ‘popular’ or ‘fraud’. We also allow a few edges between borderline videos in both classes. These have an average number of total view counts. The neighboring of the new test video is determined using highest covariance. Next, we use MSE to determine the coefficients for the weights of each neighbor in predicting the ‘viewing durations’ of the new test video.

DeepWalk is used to predict embeddings for each video using the network for ‘viewing durations’ and also for ‘view counts’. Next, we concatenate the embeddings for each video resulting in a heterogeneous embedding. Now we can predict the heterogeneous embedding for the new test video as a weighted sum over the neighborhood and the coefficients determined using the original ‘viewing duration’. Finally, we can cluster the heterogeneous embeddings into two clusters corresponding to ‘fraud’ and ‘popular’ videos. The new test video is labelled based on the closest centroid to the predicted embedding.

5. Experiments

In this section, the proposed HEAD algorithm (available on GitHub[1]) was applied to two online behavior classification problems in order to assess its efficacy. The first dataset was provided by an online advertisement company. A sequence of click from the same IP address that does not result in a purchase is labelled as a Fraud. The second dataset is collected from a university campus. The dataset has two types of information (1) Video viewing durations for different YouTube advertisements on campus (2) View count history for the video from YouTube API. The videos are labelled as fraud if the total view counts are below a threshold.

We compare the F-measure of the proposed HEAD with two baselines: (a) k-Nearest neighbor (b) Multinomial Naïve Bayes. The k-Nearest neighbor algorithm computes the class label for a node based on the nearest neighbors in the feature space. Since, we are predicting labels using the neighbors in a social network hence it is ideal to compare with this method. The multinomial Naïve Bayes model is a popular choice for discrete datasets. For example, in the click fraud data there are 322 possible Apps in the store. Hence, the click sequence is discrete and can be modeled as a multinomial. Lastly, we also compare the proposed model with two heterogenous network baselines on YouTube dataset: (a) Metapath and (b) HIN.

5.1. Advertisement tracking

When companies advertise online, they can become victims of click frauds. A click fraud happens when empty clicking of an advertisement drives up the cost and results in misleading click data. The current approach to prevent click fraud for an app developer is to measure the journey of a user’s clicks across their portfolio, and flag IP addresses which produce lots of clicks, but never end up installing apps. They ‘TalkingData’ challenge provides the app viewing history for each IP address resulting in a sequence of clicks. The aim is to predict if the IP address resulted in buying an app or was it just a visit or a fraud user. Table 1 provides the statistics for click fraud dataset. Here, we consider a balanced dataset of 2000 fraud and 2000 genuine click sequences. Each sequence has a length of 30 clicks and each click is an App id from 1 to 322. For testing we created another balanced dataset of 100 fraud and 100 genuine IP addresses.

Table 2 compares the F-measure of the proposed HEAD with two baselines: (a) k-Nearest neighbor (b) Multinomial Naïve Bayes. Both the baselines have only around 50% accuracy on ‘click fraud’ identification. In contrast, the proposed HEAD algorithm has an accuracy of 73%. Hence, it shows an improvement of over 20%. The multinomial model has slightly higher F-measure of 56% for ‘fraud’ click sequences compared to only 46% for human sequences. The proposed model performs equally well on both class labels. Hence, we can conclude that the proposed model is ideal for detecting fraudulent activity on the web.

Table 1
Statistics of click fraud and video advertisement fraud dataset.

Click Fraud	
# of Apps	322
# IP Address for Testing	2000 Fraud and 2000 Human
# of Click Time Stamps	30
Video Advertisement Fraud	
# of Videos	5667
# of Views	99,568
# Videos for Training	100 Fraud and 100 Normal
# Videos for Testing	50 Fraud and 50 Normal
# of historic viewing durations	50 engagements
# of historic view counts	90 days

Table 2
Comparison of F-measure of HEAD against k-NN and Multinomial models. The last row shows the F-measure of Heterogeneous model.

	Dataset	Fraud	Real	Total
Multinomial	Fraud	0.56	0.46	0.51
	Video	0.42	0.52	0.47
	Count	0.49	0.56	0.52
k-NN	Fraud	0.50	0.50	0.50
	Video	0.47	0.45	0.46
	Count	0.61	0.53	0.57
HEAD	Fraud	0.7	0.75	**0.73**
	Video	0.71	0.73	**0.72**
	Count	0.60	0.56	**0.58**
	Video+Count	0.75	0.81	**0.78**

The F-measure of HEAD shown in bold is the highest across the three algorithms for each dataset.

5.2. Video engagement

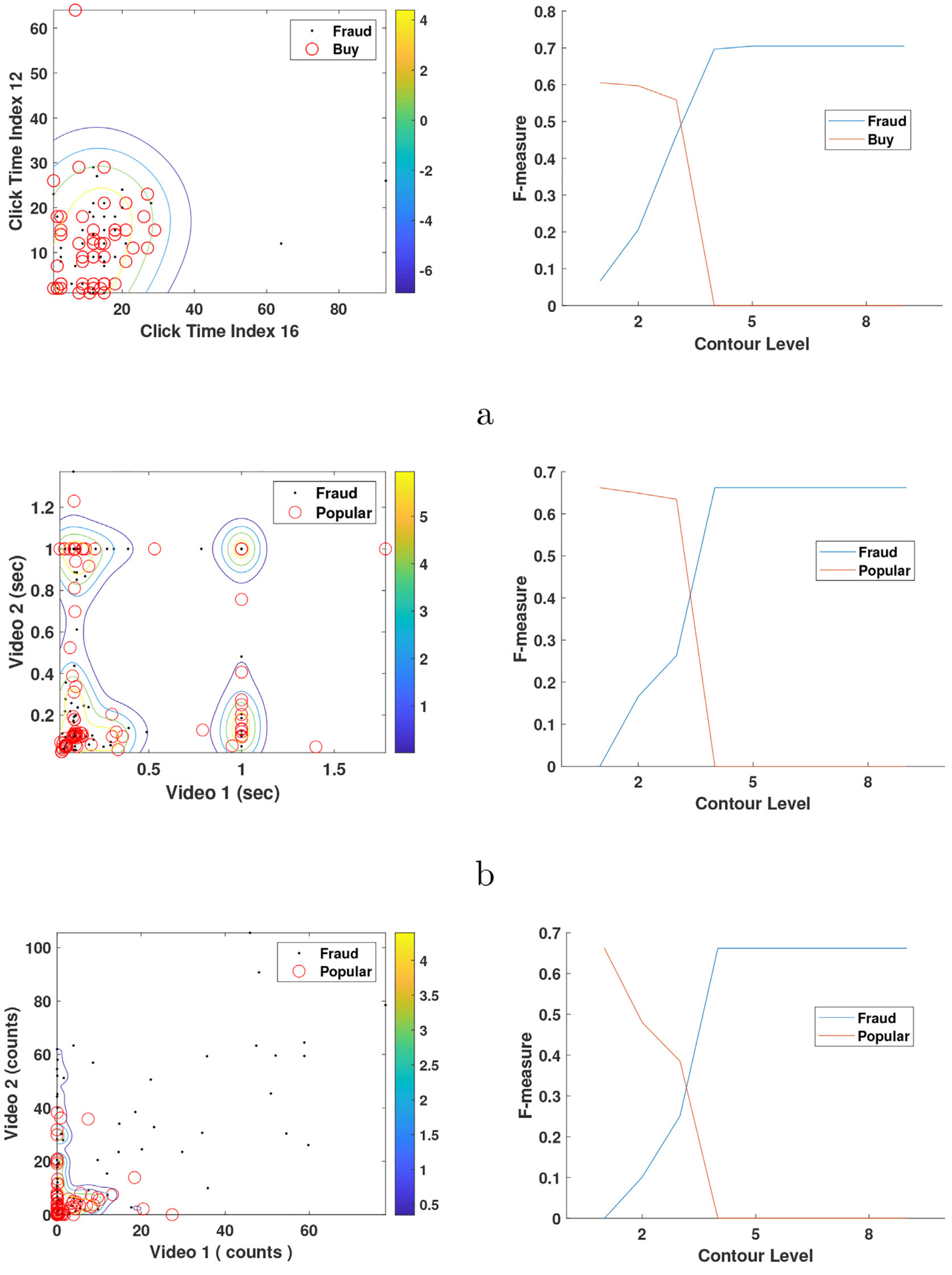
We consider video advertisement consumption of YouTube data in a university campus [2]. The skipping behavior or the number of seconds of viewing duration for each video was recorded. This was achieved using the TSTAT tool to collect HTTP requests over six months when users were exposed to video advertisements. A total of 5,668 video ads were identified and matched to 99,658 views. This data was combined with YouTube API to collect the total duration and view counts for the video. In order to study engagement, we divide the viewing duration by the total duration of each video. Next, we represent each video by a vector of normalized viewing durations for all viewers on the campus. Table 1 provides the statistics for the video fraud dataset. Here, we consider a balanced dataset of 100 fraud and 100 genuine viewing durations. Each sequence has 50 viewing durations and 90 days of historic view counts. For testing we created a similar balanced dataset of 50 fraud and 50 genuine viewing durations. As previously explained here we use the total view count to threshold and label the training videos.

For labeling the videos we use the total view counts at the end of six months. The training videos were labelled as ‘popular’ if the view counts were above a threshold and the remaining videos were labelled ‘fraud’. The YouTube API is also able to provide the evolution of a video’s popularity from the time it was uploaded

[1] http://github.com/ichaturvedi/heterogenous-engagement-auxillary-deepWalk.

until the time it is viewed. This is referred to as the view count history. Hence, we can also represent each video as a vector of view counts over time and the label is determined using the total view counts at the time of viewing.

We compute the network embeddings for videos on both types of datasets namely viewing durations and view counts. Next, we combine both of the embeddings while predicting the label for new test video as described in Section 4. Table 2 compares the F-measure of the proposed HEAD with two baselines: (a) *k*-Nearest neighbor (b) Multinomial Naïve Bayes. When using only video viewing durations both baselines show around 46% F-measure. In contrast, the proposed HEAD algorithm has an F-measure of 72%. For the historic view counts the multinomial model had an F-measure of 52% and the *k*-NN model had a much higher F-measure of 57%. Here, the proposed HEAD is only slightly higher in F-measure with 58%. This shows us that *k*-NN is suitable



a

b

c

Fig. 5. One-class SVM contours and F-measure for (a) Click Fraud (b) Video Engagement (c) View Counts.

Table 3
Comparison of F-measure of HEAD against Heterogenous network based models on YouTube dataset.

HIN [18]	Metapath [12]	HEAD
0.37	0.49	**0.78**

The F-measure of HEAD shown in bold is the highest across the three algorithms for each dataset.

model for view count data. Lastly, we consider the combined embeddings of both view durations and view counts in the heterogeneous model. Here, we see an improvement of over 6% over the homogenous video model and the F-measure is 78%. Hence, we can conclude that the proposed model is optimal for predicting video engagement.

5.3. Parameters

For the one-class SVM we had to compare a pair of videos or click time-stamps to determine the contours. Hence, we randomly choose 100 combinations from the dataset. The F-Measure is computed for each contour level and the value that gives the highest accuracy for both classes is chosen as the optimal boundary for classification. Fig. 5 shows the one-class contours for (a) Fraud advertisements dataset (b) Video Engagement and (c) View count history data. We can see that for the Fraud dataset the optimal contour value is 3 and for video data it is slightly lower at 2.5. Next, using this value we determine the outliers and discard them from the dataset before training the DeepWalk embedding. For DeepWalk we consider an embedding of length 64 for each node following previous authors.

5.4. Comparison with heterogenous network models

In [12], the authors use a Metapath based heterogeneous network for user intent recommendation in online product websites. They define a heterogeneous network where each node can be a user, an item or a query. Next, to reduce the complexity of the model they constrain the query to be made up of a set of terms whose embedding's are fixed. In effect they are transforming the heterogeneous model to a homogenous model since each node embedding is made up of the same constituents. Instead, the proposed HEAD algorithm will learn the latent embedding's for each type of dataset independently and then combine them together using network regression. Due to concatenation of embedding vectors the complexity of the model is low. Table 3 shows that Metapath algorithm performs poorly on the YouTube dataset with an F-measure of only 0.49. Here, to determine an edge between a view counts and viewing durations we consider the covariance between total view counts and viewing durations for each day.

Another application of heterogeneous information networks (HIN) is in aspect extraction from text reviews [18]. The owner of a shop can learn phrases also known as aspects that correspond to the overall rating for a product. A topic distribution is considered to model the gap between aspect ratings and overall review ratings. However, it is difficult to clearly define structural relationships between a product, a shop and a critic node. In contrast, we propose to independently learn embedding's for each node type and concatenate them for each product in latent space. Table 3 shows that HIN algorithm performs poorly on the YouTube dataset with an F-measure of only 0.37.

6. Conclusion

We propose a model that aims to fuse heterogeneous meta data such as view counts and viewing durations for classification of engagement level of a YouTube advertisement. Only a small fraction of videos is fake hence we consider a one-class model where the objective is to maximize the distance from the origin. The classification boundary takes shape of a contour around origin and videos beyond the biggest contour are discarded as outliers.

Engagement of consumers is represented as a time series of views over three months. We can also determine the sequence of videos viewed using engagement as a similarity metric. We create a network of advertisements for each type of data and concatenate the latent embeddings for each video resulting in the HEAD framework.

Clustering of embeddings can reveal the community of fraud videos. During testing we use network regression to predict embeddings for the new advertisement without the need for retraining. Experiments show that we outperform baselines in accuracy for both the fraud and popular class of videos.

CRediT authorship contribution statement

Iti Chaturvedi: Conceptualization, Writing - original draft. **Kishor Thapa:**Investigation, Validation.**Sandro Cavallari:** Software, Validation, Writing - review & editing. **Erik Cambria:** Supervision, Writing - original draft. **Roy E. Welsch:** Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is partially supported by the Computational Intelligence Lab (CIL) at the Nanyang Technological University. This work is also partially supported by the College of Science and Engineering at James Cook University.

References

[1] D. Agarwal, B.C. Chen, P. Elango, Spatio-temporal models for estimating click-through rate, in: WWW, 2009..

[2] M. Arantes, F. Figueiredo, J.M. Almeida, Understanding video-ad consumption on youtube: A measurement study on user behavior, popularity, and content properties, in: WebSci, 2016, pp. 25–34.

[3] D. Camacho, A. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, E. Cambria, The four dimensions of social network analysis: An overview of research methods, applications, and software tools, Information Fusion 63 (2020) 88–120.

[4] E. Cambria, N. Howard, J. Hsu, A. Hussain, Sentic blending: Scalable multimodal fusion for continuous interpretation of semantics and sentics, in: IEEE SSCI, Singapore, 2013, pp. 108–117.

[5] S. Cavallari, E. Cambria, H. Cai, K. Chang, V. Zheng, Embedding both finite and infinite communities on graph, IEEE Computational Intelligence Magazine 14 (2019) 39–50.

[6] I. Chaturvedi, S. Cavallari, E. Cambria, V. Zheng, Learning word vectors in deep walk using convolution, in: FLAIRS, 2017, pp. 323–328.

[7] I. Chaturvedi, R. Satapathy, S. Cavallari, E. Cambria, Fuzzy commonsense reasoning for multimodal sentiment analysis, Pattern Recognition Letters 125 (2019) 264–270.

[8] I. Chaturvedi, C. Su, R. Welsch, Fuzzy aggregated topology evolution for cognitive multi-tasks, Cognitive Computation 13 (2021) 96–107.

[9] X. Chen, J. Chen, L. Ma, J. Yao, W. Liu, J. Luo, T. Zhang, Fine-grained video attractiveness prediction using multimodal deep learning on a large real-world dataset, in: WWW, 2018, pp. 671–678.

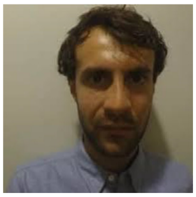
[10] X. Chen, G. Yu, J. Wang, C. Domeniconi, Z. Li, X. Zhang, Activehne: Active heterogeneous network embedding, in: IJCAI, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 2123–2129.

[11] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: KDD, 2017, pp. 135–144.
[12] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, Y. Li, Metapath-guided heterogeneous graph neural network for intent recommendation, in: KDD, 2019, pp. 2478–2486.
[13] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6 (1984) 721–741.
[14] T. Graepel, J.Q. Candela, Borchert, T., Herbrich, R., 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine, in: ICML..
[15] M. Grassi, E. Cambria, A. Hussain, F. Piazza, Sentic web: A new paradigm for managing social media affective information, Cognitive Computation 3 (2011) 480–489.
[16] Y. Hou, Z. Ma, C. Liu, C.C. Loy, Learning to steer by mimicking features from heterogeneous auxiliary networks, in: AAAI, 2019, pp. 8433–8440.
[17] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, IEEE Transactions on Neural Networks and Learning Systems 32 (2021).
[18] Y. Ji, C. Shi, F. Zhuang, P.S. Yu, Integrating topic model and heterogeneous information network for aspect mining with rating bias, in: PAKDD, 2019, pp. 160–171.
[19] M. Marciel, R. Cuevas, A. Banchs, R. González, S. Traverso, M. Ahmed, A. Azcorra, Understanding the detection of view fraud in video content portals, in: WWW, 2016, pp. 357–368.
[20] H.B. McMahan, G. Holt, D.S., et al., Ad click prediction: a view from the trenches, in: KDD, 2013..
[21] S. Mongy, A study on video viewing behavior: application to movie trailer miner, IJPEDS 22 (2007) 163–172.
[22] P. Perera, R. Nallapati, B. Xiang, OCGAN: one-class novelty detection using gans with constrained latent representations, in: CVPR, 2019, pp. 2898–2906.
[23] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: KDD, 2014, pp. 701–710.
[24] B. Perozzi, S. Skiena, Exact age prediction in social networks, 2015, pp. 91–92.
[25] C. Shi, B. Hu, W.X. Zhao, P.S. Yu, Heterogeneous information network embedding for recommendation, IEEE Transactions on Knowledge and Data Engineering 31 (2019) 357–370.
[26] L. Stappen, A. Baird, E. Cambria, B. Schuller, Sentiment analysis and topic recognition in video transcriptions, IEEE Intelligent Systems 36 (2021) 88–95.
[27] H.N. Tran, E. Cambria, A survey of graph processing on graphics processing units, The Journal of Supercomputing 74 (2018) 2086–2115.
[28] S. Venkataraman, A. Blum, D. Song, S. Sen, O. Spatscheck, Tracking dynamic sources of malicious activity at internet scale, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta, (Eds.), NIPS, 2009, pp. 1946–1954..
[29] J. Verma, S. Gupta, D. Mukherjee, T. Chakraborty, Heterogeneous edge embedding for friend recommendation, in: European Conference on Information Retrieval, Springer, 2019, pp. 172–179.
[30] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, C. Zhang, Attributed graph clustering: A deep attentional embedding approach, in: IJCAI, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 3670–3676.
[31] H. Yan, X. Chen, C. Gao, Y. Li, D. Jin, Deepapf: Deep attentive probabilistic factorization for multi-site video recommendation, in: IJCAI, 2019, pp. 1459–1465.
[32] H. Zenati, M. Romain, C. Foo, B. Lecouat, V. Chandrasekhar, Adversarially learned anomaly detection, in: ICDM, 2018, pp. 727–736.
[33] H. Zhang, X. Cao, J.K.L. Ho, T.W.S. Chow, Object-level video advertising: An optimization framework, IEEE Transactions on Industrial Informatics 13 (2017) 520–531.
[34] K. Zhang, Y. Li, J. Wang, E. Cambria, X. Li, Real-time video emotion recognition based on reinforcement learning and domain knowledge, IEEE Transactions on Circuits and Systems for Video Technology (2021).

Iti Chaturvedi completed her computer engineering at National University of Singapore on the prestigious Singapore Airlines scholarship. She also holds a PhD on predicting gene expressions using dynamic Bayesian networks from Nanyang Technological University. After her PhD, she worked with Temasek Labs, Singapore to understand the mood of consumers about political events as well as products. She is an active reviewer for grants and publications in the areas of NLP and image processing. Currently, she is a lecturer in the discipline of Information Technology at James Cook University, Townsville.

Roy Welsch Professor of Management Science and Statistics at the Massachusetts Institute of Technology Sloan School of Management and the MIT Statistics and Data Science Center. My Ph.D. is in Mathematics from Stanford University. I am currently working on using NLP and sentiment analysis to find robust financial models, machine learning to repurpose existing drugs for new uses, and univariate flagging algorithms to improve the transparency of data science models.

Erik Cambria is the Founder of SenticNet, a Singapore-based company offering B2B sentiment analysis services, and an Associate Professor at NTU, where he also holds the appointment of Provost Chair in Computer Science and Engineering. Prior to joining NTU, he worked at Microsoft Research Asia (Beijing) and HP Labs India (Bangalore) and earned his PhD through a joint programme between the University of Stirling and MIT Media Lab. His research focuses on the ensemble application of symbolic and subsymbolic AI to natural language processing tasks such as sentiment analysis, dialogue systems, and financial forecasting. Erik is recipient of many awards, e.g., the 2019 IEEE Outstanding Early Career Award, he was listed among the 2018 AI's 10 to Watch, and was featured in Forbes as one of the 5 People Building Our AI Future. He is Associate Editor of several top AI journals, e.g., INFFUS, IEEE CIM, and KBS, Special Content Editor of FGCS, Department Editor of IEEE Intelligent Systems, and is involved in many international conferences as program chair and invited speaker.



Sandro Cavallari currently working as a research scientist in the innovation lab of PayPal. I'm currently the leader of a few projects related to NLP and graph embedding in support of customer service and risk assessment. Currently, my research interest focus on multi-language and zero-shot approaches for NLP. These are some of the most important challenges that prevent NLP to be broadly applied in the industry. My PhD was on predicting community embedding's in Graphs from Nanyang Technological University, Singapore.

Kishor Thapa is a PhD candidate at James Cook University. He has a Masters in Engineering from University of Technology, Sydney. His current research involves development of computational tools towards efficient monitoring and sustainable management of groundwater resources. His research focus on robust solution and management of groundwater resources in coastal aquifers.