HIN-RNN: A Graph Representation Learning Neural Network for Fraudster Group Detection With No Handcrafted Features

Saeedreza Shehnepoor®, Roberto Togneri®, Senior Member, IEEE, Wei Liu, Member, IEEE, and Mohammed Bennamoun®, Senior Member, IEEE

Abstract—Social reviews are indispensable resources for modern consumers’ decision making. To influence the reviews, for financial gains, some companies may choose to pay groups of fraudsters rather than individuals to demote or promote products and services. This is because consumers are more likely to be misled by a large amount of similar reviews, produced by a group of fraudsters. Semantic relation such as content similarity (CS) and polarity similarity is an important factor characterizing solicited group frauds. Recent approaches on fraudster group detection empirically identified features of group behaviors that failed to capture the semantic relation of review text from the reviewers. In this article, we propose the first neural approach, HIN-RNN, a heterogeneous information network (HIN) compatible recurrent neural network (RNN) for fraudster group detection that makes use of semantic similarity and requires no handcrafted features. The HIN-RNN provides a unifying architecture for representation learning of each reviewer, with the initial vector as the sum of word embeddings (SoWEs) of all review text written by the same reviewer, concatenated by the ratio of negative reviews. Given a co-review network representing reviewers who have reviewed the same items with similar ratings and the reviewers’ vector representation, a collaboration matrix is captured through the HIN-RNN training. The proposed approach is demonstrated to be effective with marked improvement over state-of-the-art approaches on both the Yelp (22% and 12% in terms of recall and F1-value, respectively) and Amazon (4% and 2% in terms of recall and F1-value, respectively) datasets.

Index Terms—Fraudster group, heterogeneous information network (HIN), HIN-recurrent neural network (RNN), sum of word embedding (SoWE).

I. INTRODUCTION

M odern consumers trust more reviews from fellow consumers who used the products or services than advertisements. For financial gain, fabricated fraud reviews are rife on social review platforms with the purpose of either promoting one’s own products or to demote rivals’ products or both. Even worse, fraudsters may form groups that collectively attack (or promote) products for different purposes such as dominance over the sentiment of a target product, or sharing the overall workload of the fraud review effort [1]–[4]. Since the first proposed approach on fraud detection [5], most studies focused on detecting individual fraudsters [3], [6]. Only recently has there been an increased research effort targeting fraudster groups [7]–[10]. Fraudster groups are more effective in misleading consumers due to the volume and consistency of their coordinated group reviews. Consequently, they do more damage to the reputation of a review platform by undermining the trust of consumers. Fraudster group detection is a much more challenging task than individual fraudster detection because, fraudster groups can distribute their suspicious behavior over all fraudsters in the group, and through camouflage so no single fraudster stands out [1], [4]. These groups can also produce batches of fake reviews with more severe damage in a short span of time, affecting the reputation of the target product.

Recent approaches for fraudster group detection are dominated by frequent itemset mining (FIM)-based algorithms [7], [10] and graph-based approaches [8], [9]. FIM-based methods generally utilize a two-step find-and-rank pipeline to first find the candidate groups and then rank them, whereas graph-based approaches rely on graph partitioning or clustering algorithms [4] for determining candidates. However, both suffer from various limitations as outlined below.

First, approaches such as Ji et al. [8], and Zhang et al. [9] use handcrafted group-level features (e.g., group size or group rating deviation) that are generally unable to capture real behaviors of many reviewers in a group, as fraudsters in a group may change strategies to avoid detection. Second, members in a group not only perform collective activities but also aim to increase the impact by maintaining consistency in their review semantics [2]. However, as consistency may not occur altogether (by all members in the group), handcrafted features such as the number of words in a review or pairwise cosine similarity [11] of contents will fail to produce a valid global representation of reviewers. Different semantic dependencies may occur simultaneously in a fraudster group such as content similarity (CS), positive semantics of written...
reviews, or negative semantics [12]–[14]. Previous research considered either the semantic relation at a global level or pair-wise level. Global-level features such as the number of words in a review can be easily manipulated by individual reviewers whereas pair-wise-level features such as cosine similarity are too fine grained. A pair of reviewers with dissimilar content or polarity should not rule out the group-level similarity.

The semantic dependencies between reviewers are demonstrated to play an important role in fraudster (single user or fraudster groups) activities [6], [11]. However, there is a lack of effective representations. Third, a group of reviewers could form a genuine group who write reviews on different items out of the same interest. This is likely to happen when different reviewers write reviews on similar items (i.e., co-reviewing), often with multiple interaction histories. A fraudster in a genuine group (i.e., a fraudster imposter) could lead to the group being detected as a fraudster group using recent approaches. In such cases, the genuine groups will be wrongly predicted as fraud. This leads to a higher false positive (FP) rate. Conversely, fraudster groups could camouflage themselves by writing some genuine reviews, to avoid detection. We refer to such reviewers (a genuine reviewer in a fraudster group or a fraudster in a genuine group) as deviant reviewers. This increases the false negative (FN) rate of the classification.

To address these challenges, we propose a novel framework with four steps.

First, we employ the sum of word embeddings (SoWEs) inspired by the theory of collective intelligence (CI) [15], [16] to use all reviews written by a reviewer as features to overcome the limitations of handcrafted features. The SoWEs has been verified on various occasions to surpass more complex document embeddings [17] as a simple yet effective representation for documents. In this research, we, therefore, choose to use the SoWE of all tokens in a review as a review’s vector representation. A reviewer’s vector representation is thus the SoWE of all reviews written by this reviewer and a group’s vector representation is thus the average of vector representations of all reviewers in that group. The SoWE is further fine-tuned by training a convolutional neural network (CNN) on the fraudster detection task. A CNN is chosen in preference to recurrent neural networks (RNNs) because CNNs are better in dealing with the potential multiple aspects covered in each review [18], [19]. Although the SoWE is simple and shows a desirable performance (see Sections IV-C and IV-C2), it can be replaced with any other advanced natural language processing (NLP) technique in the proposed framework. To this end, we also use bidirectional encoder representations from transformers (BERTs) [20] for the first time in fraudster group detection to show the performance of this recent NLP technique employed for the embedding.

Next, we determine the candidate groups based on the time interval between written reviews. Fraudsters in a group complete their reviews in a relatively shorter time window compared to genuine reviewers. Previous studies estimated the time interval to be 28 days [1], [8], [21]. We use the same time-interval (i.e., 28 days) to determine the possible collaboration between reviewers. Hence, every two reviewers co-reviewing at least two items with the same rating form a connection (not necessarily in terms of fraudster activity). The outputs of this step are subgraphs that capture possible collusion between reviewers. In the third step, we propose the HIN-RNN, i.e., heterogeneous information network (HIN) compatible RNN for the fraudster group detection task. The intuition behind using the HIN-RNN is twofold: first, the long-range dependency between the reviewers is captured through the RNN’s autoregressive model, and second, the reviewer dependency modeling is further improved by incorporating prior knowledge of a reviewers’ type such that heterogeneous nodes are allowed in the network. This results in a better dependency prediction on different datasets (see Section IV-C3). The HIN-RNN model takes the vector representations of reviewers from the first step and the resulting subgraphs from the second step to encode the nonlocal semantic dependencies (long-range relationships) between reviewers through an autoregressive model. The HIN-RNN model addresses the limitations of another graph representation model, i.e., GraphRNN [22], which generates subgraphs with only nodes of the same type. The output of this step is a collaboration matrix indicating the collaboration between the reviewers.

Finally, given the collaboration matrix of groups, we exclude those reviewers with minimum connections, which are genuine reviewers unintentionally contributing to a fraud activity (thereby reducing FP), or fraudsters trying to camouflage to escape detection (thereby reducing FN). Then an average of the remaining reviewers’ representations is fed to a simple fully connected layer to label (genuine/fraudster) each group. We can summarize our contributions as follows.

1) We extend RNN to support the representation learning of different node types in a graph, termed the HIN-RNN. Our entire proposed approach showed an average increase of 22%, and 12% for recall and F1-value, respectively, on the fraudster group detection task over most recent methods on the Yelp dataset (see Section IV-C1). It is noted that although the performance of the HIN-RNN is measured on the fraudster group detection task with only two types of reviewers involved (fraudster and genuine), the HIN-RNN can be applied to a broad range of graph generation tasks with multiple node types in the graph (see Section IV-C3).

2) Another contribution is that we devised a learned vector representation of reviewers, without any handcrafted featured engineering, purely based on the review texts reviewers have written. To compare performance, we have tried both simple Word2Vec [23] and the contextual word embeddings (WEs) BERT [20]. Employing the two-step approach shows a boost in performance by 12%, and 7% for recall and F1-value on Yelp, respectively, compared to the handcrafted features employed by Ji et al. [8] (see Section IV-C2).

3) We use the collaboration matrix to exclude reviewers with minimum connections (deviant reviewers) from a group. The exclusion of such deviant reviewers improves the performance of the proposed approach on Yelp (Amazon) substantially by 7% (8%), 12% (7%), and 9% (7%) in terms of precision, recall, and F1-value, respectively (see Section IV-C4).
The rest of the article is structured as follows. In Section II, we present the related work. In Section III, we introduce our methodology. In Section IV, we provide the experimental evaluation. We conclude the article with an outlook to future work in Section V.

II. RELATED WORKS

In this section, we discuss the recent studies; first on the representation learning for fraud detection, then we explore the most recent studies on fraudster group detection.

A. Representation Extraction

Recent advances in NLP research have shown promising results on different tasks such as text classification [20]. However, few studies investigated such techniques in fraud detection. Heidari and Jones [24] proposed an approach to analyze the sentiment of the posts on Twitter and acquire a score for each post of a user in the platform. Then topic-independent sentiment features were extracted: number of neutral, positive, and negative tweets for a user, the sum of positive and negative scores of the user, and then the average value for the positive and negative scores of the user. Such features were then fed to a simple neural network to score the positive and negative scores of the user, and then the average neutral, positive, and negative tweets for a user, the sum of independent sentiment features were extracted: number of pages visited (e.g., Homepage → List → Detail → ···). The results showed an accuracy of 94% on the dataset.

Li et al. [26] investigated click fraud detection, referring to a type of Internet advertising strategy in which an advertiser pays a publisher when the ad is clicked. Three types of features were used. First, statistical features such as the absolute position of an ad on a website, the id of the advertiser, timestamp of a given click, and so on are extracted. Then behavioral features were extracted in terms of the sequence of pages visited (e.g., Homepage → List → Detail → ···) and fed to BERT to convert it to a vector representation. Finally, a homogeneous network is modeled, where each node represents a device, if two devices used the same internet protocol (IP), they are connected. Network-embedding techniques are used to obtain a medium embedding for the ad. These features were then concatenated and fed to a fully connected layer to determine whether an ad is a fraud or not. The results showed an improvement of 7.2% on area under curve (AUC) on a dataset collected from Alibaba.

Shishah [27] employed BERT [20] to extract a representation for articles in the Politifact [28] and Pymedia [29] datasets, both including news articles collected from the websites. The representation for each word vector was then concatenated and fed to multilayer perceptron to determine whether the article is fake news or genuine. The proposed approach showed an accuracy of 0.83 and 0.84 on the Pymedia and Politifact datasets, respectively.

In this study, we rely on SoWE as the feature extraction technique due to its: 1) simplicity and 2) promising results (see Sections IV-C and IV-C2). In addition, to compare the performance of the system on different embedding techniques, we conducted experiments with BERT as another representation learning technique and compared it against SoWE in Section IV-C2.

B. Fraudster Group Detection

Studies on fraudster group detection fall under two groups, FIM or graph-based approaches.

1) Frequent Itemset Mining: FIM refers to approaches that assume members with the same set of reviewed items (itemset) as a candidate group [30]. Such an assumption can also be extended to other aspects of reviewers’ relations in different studies. Allahbakhsh et al. [7] propose a data model where two types of reviewers and items connect via rating activities. Fraudster groups are detected by a biclique detector. A biclique refers to a group of reviewers who not only write reviews (FIM basic assumption) but also rate similarly the same group of items. After determining the candidate groups using the biclique criteria, different handcrafted fraudster group activity indicators [i.e., group-rating value similarity (GVS), group-rating time similarity (GTS)] are used to measure the potential of a group being fraudulent. A scoring function was applied to integrate these group features to obtain the final scores. Xu et al. [10] chose FIM to generate the candidate groups at least two reviewers with at least three co-reviewed businesses. A homogeneity-based-collusive behavior measure (h-CBM) is then proposed based on the targeted item, rating, temporal traits, and reviewer activity. The groups are scored through an unsupervised scoring model called latent collusive model (LCM). In a nutshell, FIM approaches come with different shortcomings such as overlooking the deviant reviewers (i.e., single genuine reviewer in fraudster group and vice-versa) and the importance of the semantic connection between reviewers in a group [7], [10].

2) Graph-Based: Graph-based fraudster group detection utilizes graph-based methods such as graph partitioning or clustering algorithms to address FIM model’s limitations.

Ji et al. [8] propose a method where products are the focal points to overcome the limitations of FIM’s reviewer-focused approaches. First, eight handcrafted group-level features (group rating deviation, group size, group review tightness, group one day reviews, group extreme rating ratio, group co-activeness, and group co-active review ratio), six individual fraudster features (ratio of extreme rating, rating deviation, the most reviews one-day, review time interval, account duration, and active time interval reviews), and three product-related features (product rating distribution, product average rating distribution, and suspicious score) are extracted. Although such handcrafted features have been broadly employed for the fraud detection task, they could easily be manipulated by fraudsters or fraudster groups to fool a detection system. For example, fraudster groups can be separated into smaller groups to avoid detection [9]. Some features can also be misleading. For example, the ratio of extreme rating, and rating deviation, can result in different indications; if the ratio of extreme rating increases for a group, the deviation is decreased which neutralizes the other. Targeted items are then identified based on these item-related features. A kernel density estimation (KDE) method is used to calculate the burstiness in items. No review semantics is considered in identifying the connection between the reviewers.

Zhang et al. [9] propose a three step framework to address the limitations of the FIM model. In the first step, a graph
of reviewers (nodes) is constructed based on the different relations (edges) such as correlation and relevance between the reviewers’ ratings and their review times. In the next step, a label propagation algorithm based on propagation intensity and an automatic filtering mechanism is applied to obtain the final labels. Graph-based methods are also commonly used in other spam detection tasks. As one of the most recent approaches, Li et al. [26] proposed a large-scale anti-spam method based on graph convolutional network (GCN) to detect spam advertisements, called GCN-based anti-spam (GAS). The embedding of each review, reviewer, and item is captured through concatenating the texts for each component based on the pretrained skip-gram word2vec technique by Mikolov et al. [31]. The embeddings are then concatenated and fed to a GCN to determine the probability of an ad being spam or not. A comparison of FIM-based and graph-based approaches is displayed in Fig. 1.

3) Research Gap: Previous approaches in both FIM and the graph-based methods extract handcrafted features which could be easily manipulated to avoid the detection approaches [32], [33]. On the other hand, WEs of review text written, have been shown to contain more contextual embedding information than handcrafted features [19]. We recognize this in this research and use WEs refined through a CNN as reviewer representations. Also, the semantic similarities between reviews from reviewers of a group are overlooked by the FIM and the graph-based approaches. In this research, we, therefore, devise a new graph representation learning neural network, to: 1) model temporal relationship (co-reviewing) between reviewers to predict the collaboration matrix; 2) to introduce the semantic collaboration through RNN; and 3) current approaches treat the collaboration networks as static objects and apply no network pruning or finetuning. We recognize that a fraudster could camouflage in a genuine group and vice-versa, so pruning the least connected node helps mitigate this problem.

III. PROPOSED APPROACH

A. Problem Definition

Assume a set of review text \((T)\) written by a set of reviewers \((R)\), for a set of items \((I)\), with ratings (rating), the goal is to first, determine candidate groups \((G = \{g_1, \ldots, g_i, \ldots, g_m\})\) where \(g_i \subseteq R\), and then classify each group into \(L_G = \{\text{fraudster, genuine}\}\).

In this research, we take advantage of the obviously and abundantly available but often overlooked, review text as surrogates for reviewers, and establish relations between reviewers based on semantic similarities of their review text. Our proposed methodology, therefore, focuses on the meaningful vector representation of reviewers and their relations obtained through four main modules as illustrated in Fig. 2. A representation is obtained for each reviewer through aggregating written reviews in module 1, while initial candidate groups are created through co-review assumptions in module 2. The two vector representations are then combined using the HIN-RNN module to obtain a refined collaboration matrix. Finally, we exclude the deviant reviewers in a group to apply a simple fully connected layer to the average of the remaining reviewers’ representation to output the final labels.

B. Reviewer Representation Extraction

CI [15], [16] states that the online footprints of a person provide an important interpretation of his/her behavior. For example, a recruiter may look through a person’s reviews online to understand his/her general behavior. Therefore, a natural representation of a reviewer could be the collection of reviews. Traditional text-based features (e.g., the number of capital words in a review text, review text length) fail to represent a user and can easily be manipulated by fraudsters to fool the detectors. In contrast, WE techniques have shown significant performance in fraud detection [19]. This is largely attributed to WE’s capabilities in capturing semantic similarities. In other words, review texts do not have to contain the same tokens to be deemed as similar by WE techniques. SoWEs refers to an operation (in this case, addition) that aggregates WEs to produce a high-level description of a sentence or a reviewer. Therefore, we use the SoWE at two different levels in this work: reviewer level and group level. It is possible to use either sentence or document embeddings techniques such as sec2vec [34] and skip-thought [35], we choose SoWE due to its simplicity and proven efficiency in various domains [19], [36], [37]. The sentence-level embeddings are finetuned by a CNN with max-pooling. These aggregated representations are much more meaningful covering the global characteristics of each reviewer compared to handcrafted features as demonstrated in Section IV-C2. Extracting a reviewer-level representation follows three main steps of aggregations to cover the global characteristics of a reviewer.

1) Word Representation: After aggregation of a reviewers’ review texts, the reviews are broken down into sentences. Each sentence consists of words. In this step, each word in the sentence is initialized with a pretrained WE represented as \(e_{w_i} \in R^D\), with \(D\) as the word vector dimension.
2) Sentence Representation: CNNs are capable of capturing the global characteristics of a sentence through three different layers [17]. Here, a CNN is applied to refine the representations based on the reviewer types (genuine or fraudster). The CNN captures the reviewer-type information through reviewer labels. So the CNN is fed with the WE and trained using the labels for each reviewer. The structure of the CNN we employed is given in Fig. 3. Since most reviews are less than 400 words, we selected the reviews with less than 400 words. Then we pad words with “END,” so they can have a length of 400. In the first layer, a convolutional layer is utilized to operate as a trigram language model. The output of the first layer is given by

\[ H_i = W I_{3,i} + b. \]  

(1)

In (1), \( W \in \mathbb{R}^{1,3} \) is the weight matrix of the convolutional layer with stride 1, \( I_{3,i} \in \mathbb{R}^{1,D} \) is the concatenation of three consecutive words (as shown to be effective in fraud review detection in different studies [19], [23]) in the trigram model, and \( b \) is the bias. Next, an average pooling is applied to the mean over the words of a sentence. Finally, a tanh function is applied as an activation function. It is noted that tanh is preferred over other activation functions for the given task, as the average of tanh is around zero, thus facilitating the training process and improving the convergence. The activation function outputs the final embedding of a sentence

\[ e_s = \text{tanh} \left( \frac{1}{n-2} \sum_{i=1}^{n-2} H_i \right) \]  

(2)

In (2), \( n \) is the number of words in the sentence.

3) Reviewer Representation: The reviewer representation is obtained by first concatenating the reviewer’s sentence representations as a 2-D tensor. Next, a max-pooling layer is applied to obtain the reviewer embedding, which is then fed to a fully connected layer to label the reviewer as a fraudster or genuine. The output of this step is the reviewer embedding for each reviewer (refer to Fig. 3). To train the CNN, we used a cross-entropy as the cost function. Next, the embedding is concatenated with the negative ratio (NR) [11]

\[ NR = \frac{N(\text{rating} = 1, 2)}{N} \]  

(3)
where $N$ (rating) is the number of reviews with specific ratings (rating) in the range of 1–5 (5 is the highest) and $N$ is the total number of reviews for each reviewer. The NR is chosen because fraudsters have a higher tendency to write negative reviews. As an example, in the Yelp dataset, for every three positive reviews, there is one negative review, written by fraudsters. This ratio is from 8 (positive) to 1 (negative) for genuine reviewers. The concatenation forms the final vector representation of reviewer $r$ as $v_r$.

**C. Candidate Groups**

Fraudsters in a group tend to write their reviews in a shorter time interval than genuine reviewers in an attempt to increase the collective impact. We assume a time interval of 28 days similar to previous studies [1], [21]. Based on this assumption, every two people who wrote reviews on a set of a minimum of two common items with the same rating in the specified time-window (in this case, 28 days) form a possible collaboration [8], [9], [38]. The remaining reviewers are treated as individual reviewers and are ignored for fraudster group detection. The output of this step are subgraphs of possible groups, which we termed as the co-review networks.

**Definition:** A co-review network is a subgraph $G(R, E)$ where $R$ is a set of reviewers, each reviewer ($r \in R$) as a node connected via an undirected edge ($e \in E$) with another reviewer if they have co-reviewed, and thus represent a possible collaboration.

**D. HIN-RNN Framework**

As explained in the Introduction, fraudsters in a group develop a “deep” semantic relationship to cover the traces of the collaboration. To capture such relationships between two reviewers, the proposed model should be able to consider a long-range relationship between the reviewers to improve the detection of candidate groups. Given the efficiency of an RNN in modeling long range dependencies, we adopted an RNN to model the relationship between reviewers in a group. Our co-review networks contain two types of nodes, genuine reviewers, and fraudsters, thus, strictly speaking, an HIN or more precisely, in this case, a bipartite graph of two types of nodes. To facilitate the representation learning in our co-review network, we propose HIN-RNN, an RNN compatible with HIN. This is similar to GraphRNN [22], which uses an autoregressive model to generate graphs by training on a sequence of subgraphs. GraphRNN is capable of capturing the long-term relationships between the nodes in a graph. However, it does not support nodes of different types. Such a limitation results in the ineffectiveness of the GraphRNN in modeling the heterogeneity in a network. Our HIN-RNN addresses such limitations by utilizing the reviewer representation to better predict the adjacency matrix of the nodes. Although such changes add complexities to the network, this in turn, improves the performance, both theoretically and experimentally.

HIN-RNN brings the merits of both HIN and RNN to support the different types of nodes with long-range dependencies, suitable for fraudster group detection task: From a technical perspective, employing the representation of reviewers to model the long-range dependency provides a preknowledge of how the reviewer is related to other reviewers in a group. With preknowledge given for each reviewer, a better prediction is more likely. On the other hand, the RNN utilizes the representation and the collaboration matrix to predict the relations, used to encode the connection between reviewers of different types. This helps the HIN-RNN to a further improvement, where it better models the camouflage of fraudsters in a genuine group and unintentional contribution of genuine reviewers in a fraudster group. On the contrary, homogenous modeling of reviewers in a group (practiced by previous studies on fraudster group detection) results in a shortcoming in considering different types of reviewers, thus overlooking: 1) the useful preknowledge for each reviewer and 2) the camouflage, thus increasing the misclassification rate (see Sections IV-C4 and IV-C3).

1) GraphRNN: Adjacency matrix is a standard way of representing a graph. Here we use a node ordering $\pi$ that maps the nodes of a subgraph (group) to the rows and columns of an adjacency matrix where $r_i$ is a reviewer node. From here on, we refer to this as the collaboration matrix, where each element indicates if there is a relation between two reviewers in a group or not. Let $\Pi$ ($\Pi : R \rightarrow S$) be a set of all possible permutations of a sequence of the reviewers’ collaborations, where $R$ is a set of reviewers and $S$ is an ordered sequence $\{\pi^{-1}(1), \ldots, \pi^{-1}(n)\}$. $\Pi^{-1}(i)$ is the inverse function of $\pi$ that maps a sequence position to a node. For clarity let us use $r^\pi_i$ to represent the $i$th reviewer according to order $\pi\{\pi^{-1}(1), \ldots, \pi^{-1}(i), \ldots, \pi^{-1}(n)\}$. An illustrated example of collaboration matrix is given in Fig. 4. Assuming all reviewers are involved then the size of $\Pi$ is $n!$. A co-review network can be modeled as a specific ordering with $\pi \in \Pi$ the corresponding collaboration matrix represented as $A^\pi \in \mathbb{R}^{n \times n}$ with $A_{\pi(r_i), \pi(r_j)} = 1[r_i, r_j \in E]$, where $E$ is the set of edges. The goal is to learn a set of distributions $p(G)$ over all possible groups, $G_i \in \{G_1, \ldots, G_m\}$, where $G_i$ represents a potential grouping (e.g., a fraudster group obtained from Section III-C) of reviewers within a collaboration matrix.

To map the problem to an autoregressive model, we need to define a mapping function: $f_\pi$ from a co-review network to a sequence. For a group $G \sim p(G)$ with $n$ different reviewers and $\pi$ as the ordering (a possible collaboration between reviewers), the set of collaboration sequences for each reviewer is defined as below

$$S^\pi = f_\pi(G, \pi) = (S^\pi_1, \ldots, S^\pi_i, \ldots, S^\pi_n)$$

![Fig. 4. Illustrated example of the collaboration matrix.](image-url)
where $S^\pi_i \in \{0, 1\}^{i-1}, i \in \{1, \ldots, n\}$ is a vector of length $i - 1$, representing a collaboration sequence for $i$th reviewer $(S^\pi_i = (A_{\pi(i)}, \pi(i)), \ldots, A_{\pi(i)}, \pi(i))^T, \forall i \in \{2, \ldots, n\}$) between reviewer $r^\pi_i$ and all previous reviewers $r, j \in \{1, \ldots, i - 1\}$ in a group. In other words, $S^\pi$ provides a unique group representation $G$, where $f_G(S^\pi) = G$ ($f_G$ is the inverse of $f_S$). So, instead of learning $p(G)$, whose feature space is not easily characterized, the model learns $p(S^\pi|G)$, which is observable using the auxiliary function $\pi$. So, $p(G)$ can be represented as a joint distribution $p(G, S^\pi)$

$$p(G) = \sum_{S^\pi} p(S^\pi) \mathbf{1}[f_G(S^\pi) = G].$$

(5)

We are able to rewrite the $p(S^\pi)$ as an autoregressive conditional distribution

$$p(S^\pi) = \prod_{i=1}^{n+1} p(S^\pi_i | S^\pi_{i-1}, \ldots, S^\pi_1) = \prod_{i=1}^{n+1} p(S^\pi_i | S^\pi_{i-1}).$$

(6)

This means that the collaboration vector for each reviewer is dependent on the collaboration vectors on the previous reviewers.

2) HIN-RNN: So far, the group generation process is modeled through the mapping of the possible collaborations into a collaboration vector for each reviewer. The collaboration vector for each reviewer is modeled through an autoregressive model which conditions each reviewer’s collaboration on all reviewers in a group. The GraphRNN learns a collaboration vector for each reviewer, but it is incapable of handling multiple reviewer types. In this research, to include the reviewer types in generating the collaboration matrix, we condition the generation process on both the current reviewer’s vector representation ($v_i$ from Section III-B) and the collaboration vectors of previous reviewers

$$p(S^\pi_i | S^\pi_{i-1}) = \prod_{j=1}^{i-1} p(S^\pi_{i,j} | v_i, S^\pi_{i-1}, S^\pi_{i,j}).$$

(7)

In (7), $S^\pi_{i,j}$ is 1 if there is a collaboration between reviewer $i, j$. The structure of the HIN-RNN model is displayed in Fig. 5. Two models are required to the first to parameterize the state of the collaboration matrix, and the second to how previous reviewers are interconnected to the current reviewer. For each model, we use a RNN to capture the distributions

$$h_i = f_1(h_{i-1}, S^\pi_{i-1}, v_i)$$

(8)

where $h_i$ encodes the state of the groups (reviewers plus their collaboration matrix) up to reviewer $i$ and $f_1$ is the function learned through training another RNN. Next, we obtain the collaboration matrix of the current reviewer

$$S^\pi_i = f_2(h_i)$$

(9)

where $S^\pi_i$ encodes the collaboration matrix obtained from the function $f_2$ using the RNN.

3) HIN-RNN: Challenges and Variants: Various challenges are introduced when combining the HIN with an RNN. The first challenge is on how to combine the representations with an adjacency matrix to support the heterogeneity of different nodes in an autoregressive model, strictly speaking, an RNN. In this study, we concatenated the representations with the collaboration matrix to obtain the collaboration matrix of the group. Alternatively, the autoregressive model can only be used to predict the representation of the next reviewer as another variant of the HIN-RNN

$$v_i = f_1(h_{i-1}, S^\pi_{i-1}).$$

(10)

This is similar to $f_1$ in (8). However, instead of using $v_i$ alongside $h_{i-1}, S^\pi_{i-1}$, the $f_1$ predicts $v_i$, given $h_{i-1}, S^\pi_{i-1}$ as the inputs. The acquired representation is then fed to a multilayer perceptron to obtain the final collaboration matrix

$$S^\pi_i = f_2(v_i).$$

(11)

This is also similar to (9). However, in this model, the HIN-RNN only uses the representation of the first reviewer to
predict the collaboration matrix of the next reviewer. Accordingly, this model is suitable for the cold-start problem [23], where there is no history of a new reviewer to acquire a preknowledge of the reviewer.

Another challenge is the WE technique used to generate the reviewer’s representation. Such a representation can be generated through different state-of-the-art WE techniques such as BERT [20], XLNet [39], and so on. Here we used SoWE, as it has shown promising results in fraud detection [4]. We compare the performance of the HIN-RNN described in Section III-D, we exclude the SoWE of BERT to demonstrate the effectiveness of the proposed approach to extract the deep semantics.

E. Group Classification

Our proposed group classification method follows three simple steps: deviant reviewer removal, group-level representation, and fully connected layer.

1) Deviant Reviewer Removal: As explained, people in a group develop co-occurring semantic relations. Given such relations encoded through the HIN-RNN, the reviewers with minimum connections are considered deviant reviewers. As such, in this step, from the groups generated at the output of the HIN-RNN described in Section III-D, we exclude the reviewers with a minimum collaboration in a group to remove the possibility of unintentional contribution from genuine reviewers in a fraud activity (reducing FP) and avoid the effects of fraudulent activity in a genuine group (reducing FN). The final output of this step includes groups with deviant reviewers excluded.

2) Group-Level Representation: Next, we extract a group-level representation (refer to Section III-B) using the CI theory. Finally, we perform an element-wise average over the representations of the remaining reviewers to obtain a final representation of each group. The element-wise average of a group’s representation is the output of this step.

3) Fully-Connected Layer: In this step, a fully connected layer is trained on the representation obtained from Section III-E2.

The proposed approach is presented in Algorithm 1.

| Algorithm 1: HIN-RNN Algorithm |
|--------------------------------|
| **Output:** The label of each group being fraudster or genuine; |
| **Input:** $T$ review texts of $R$ reviewers and ratings $rate$, written on $I$ items; |
| **Step 1:** reviewer representation; |
| **for** $r \leftarrow 1$ to $R$ **do** |
| **# Aggregate review texts of reviewer $r$;** |
| $t_r \leftarrow aggregate[T_1, T_2, \ldots, T_m]$; |
| **# Tokenize $t_r$ to $S$ sentences;** |
| $\{s_1, s_2, \ldots, s_n\} \leftarrow tokenize(t_r)$; |
| **# Sentence Representation;** |
| **for** $s \leftarrow 1$ to $S$ **do** |
| **# Tokenize $S_s$ to $n$ words;** |
| $\{w_1, w_2, \ldots, w_n\} \leftarrow tokenize(S_s)$; |
| **# Word embeddings;** |
| $[e_{w_1}, e_{w_2}, \ldots, e_{w_n}] \leftarrow WE((w_1, w_2, \ldots, w_n))$; |
| **# The sentence representation;** |
| $e_s \leftarrow CNN((e_{w_1}, e_{w_2}, \ldots, e_{w_n}))$; |
| **# Negative Ratio of reviewer $r$;** |
| $NR_r \leftarrow NR(rate_1, rate_2, \ldots, rate_n)$; |
| **# Final representation of $r$;** |
| $v_r \leftarrow maxPool(concat(e_s, \forall s \in S)) \oplus NR_r$; |
| **Step 2:** candidate groups subgraphs; |
| **if** $r_3, r_y \ co-review same item $i \in I, \forall r_3, r_y \in R$ **then** |
| **# Link the possible collaborating reviewers;** |
| $E(x, y) = 1$; |
| **Step 3:** final collaboration matrix using HIN-RNN; |
| **for** $i \leftarrow 1$ to $R$ **do** |
| $h_r_i \leftarrow RNN_i(h_{r_i-1}, E(0 : i = 1, 0 : i = 1), v_i)$; |
| **# Collaboration matrix of reviewer $i$;** |
| $G \leftarrow RNN_2(h_r)$; |
| **Step 4:** group classification; |
| **for** $g \in G$ **do** |
| $g \leftarrow \{\text{reviewer with minimum connections}\}$; |
| $v_g \leftarrow mean(v_r)$ for $r \in g$; |
| $label(g) \leftarrow fc(v_g)$; |

IV. EXPERIMENTAL EVALUATION

We compare the proposed approach to the state-of-the-art approaches to demonstrate the effectiveness of each of our innovations.

A. Experimental Setup

We used a 100-dimension continuous bag of words (CBoW) due to its effectiveness in fraud review detection [17] with a window size of 2 and batch size of 256. For training the CNN, the learning rate was $10^{-4}$, the training epochs was 30, and cross-entropy was used as the objective function. To train the group generator model, we used two RNNs with gated recurrent units (GRUs), one learns the hidden state of the group with a hidden size of 128, and the other learns the reviewer collaboration with a hidden size of 16. The two RNNs were trained jointly with a learning rate of 0.003 and in 3000 epochs.

| TABLE I |
| LIST OF DATASETS USED IN OUR CURRENT STUDY |
|----------|----------|----------|----------------|----------------|
| Dataset | reviewers | Items | Reviews | Candidate Groups |
| Yelp    | 260.277   | 3.044  | 608.596 | 9,952          |
| Amazon  | 42,655    | 6,822  | 53,777  | 2,194           |

B. Datasets

Previous studies evaluated the performance of the proposed approach on either the Yelp or Amazon dataset. To demonstrate the scalability of the proposed approach, we used both datasets. The Yelp dataset includes reviews from “20-Oct-2004” to “10-Jan-2015” and provides the labels (genuine or fraudster) for possible groups [9]. The Amazon dataset contains reviews from “01-Feb-2000” to “10-Oct-2010,” and groups are determined and labeled, similar to the Yelp dataset [8], [9]. In both datasets, the fraudsters who have...
co-reviewed the same set of items with others form fraudster groups. It is noted that the generated graphs from the HIN-RNN might be different from the ground truth, and the HIN-RNN is proposed to improve the accuracy of generated graphs (in terms of node and edge prediction). Additionally, the ground truth and the generated graphs might include fraudsters and genuine reviewers in the same group. The proposed step in Section III-E1 is to remove the reviewers from groups with the opposite label. Datasets include item id, reviewer id, the review text, the rating given by each review, and the date of the written review as the metadata. The details on the datasets are provided in Table I. For the evaluation, we used 80% of the data for the training set and 20% of the data for the test set.

C. Main Results
The performance of our proposed approach is measured using three metrics: precision \((TP / TP + FP)\), recall \((TP / TP + FN)\), and F1-value \((2 \times precision \times recall / precision + recall)\).

1) Comparison With Baselines: The performance of the proposed approach is compared against two FIM-based and graph-based systems in Table II. We also devised two different network configurations to compare the performance of the HIN-RNN with other graph generation techniques.

   a) GraphRNN + SoWE + NR: In this configuration, the reviewer representation (SoWE + NR) is not employed to refine the groups’ structure. Though, the reviewer representation is used to obtain the group-level representation.

   b) HIN-RNN + CS + Group polarity (GP): In this configuration, we utilize the HIN-RNN to incorporate the CS through calculating the average cosine similarity of the SoWE of all reviewers in a group, and also the average polarity (positive/negative) of the reviews written by the reviewers.

   c) HIN-RNN + (SoWE + NR): This is the regular proposed configuration where features are employed to refine the group structure.

The results demonstrate that the proposed approach significantly outperforms the baselines in terms of recall and F1-value on the Yelp dataset. The proposed approach also improves the baselines’ performance in terms of Precision and F1-value. The proposed approaches by Allahbakhsh et al. [7] and Xu et al. [10] are FIM-based approaches, ignoring the dependency of the reviewers in a group. Such a dependency can significantly affect the performance of the proposed approaches. On the other hand, graph-based approaches incorporate the dependency between reviewers by mapping the relations of the reviewers to graphs. As mentioned in Section II, Ji et al. [8] provide a strategy to overcome the limitations of previous works in excluding genuine reviewers who unintentionally assigned to a fraudster group. However, it still suffered from a limitation in handling such situations. So FN is not reduced significantly (i.e., recall is low) compared to Zhang et al. [9]. On the other hand, the approach by Ji et al. [8] performs better when faced with single fraudsters who co-review the nontargeted items with groups of genuine reviewers, and thus decreases FP (and increases precision). The GraphRNN, although improving the performance, still suffers from overlooking the semantic relationship between reviewers in a group. As shown the HIN-RNN still performs better than the baselines on the Yelp dataset. The reason behind the low performance compared to the Ji et al. [8] is that CS and GP can be manipulated by the reviewers in a group and hence such handcrafted features can result in misclassifications.

The HIN-RNN effectively increases the model flexibility to identify deviant reviewers’ involvement in a group and then remove them. Hence, both FP and FN are significantly reduced, resulting in a 22% improvement in recall and 12% in F1-value on the Yelp dataset.

2) Effects of the Reviewers’ Representation Versus Handcrafted Individual Features: Ji et al. [8] incorporate six handcrafted individual features (discussed in Section II). Two versions of our proposed approach are presented here to show the effectiveness of the proposed features.

   a) HIN-RNN + six features from Ji et al. [8]: This configuration utilizes the six individual fraudster features proposed by Ji et al. as discussed in Section II. In this configuration, each reviewer is represented as a concatenation of the following handcrafted features: ratio of extreme rating, rating deviation, the most reviews’ one-day review time interval, account duration, and active time interval reviews. We then use the handcrafted features to refine the candidate groups.

   b) HIN-RNN + (SoWE + NR): This configuration utilizes SoWE as the feature representation (see Section III-B).

   c) HIN-RNN + BERT: In this configuration, we utilize BERT to encode the input review with CBoW as the WE, position embeddings, and the segment embedding. Hence, there is no discrimination between reviews with a single sentence and multiple sentences. We aggregated the reviews for each reviewer. Next, we pretrained the BERT with the unsupervised objective function. In the fine-tuning step, we used the

| Dataset | Metric | Precision | Recall | F1-value | Precision | Recall | F1-value |
|---------|--------|-----------|--------|----------|-----------|--------|----------|
| Yelp    |        | 0.62      | 0.15   | 0.25     | 0.61      | 0.18   | 0.28     |
|         |        | 0.75      | 0.50   | 0.60     | 0.65      | 0.40   | 0.50     |
|         |        | 0.70      | 0.20   | 0.32     | 0.80      | 0.45   | 0.58     |
|         |        | 0.83      | 0.60   | 0.69     | 0.82      | 0.92   | 0.86     |
|         |        | 0.77      | 0.72   | 0.75     | 0.81      | 0.88   | 0.84     |
|         |        | 0.70      | 0.66   | 0.68     | 0.75      | 0.63   | 0.68     |
|         |        | 0.81      | 0.82   | 0.81     | 0.85      | 0.90   | 0.87     |
| Amazon  |        |           |        |          |           |        |          |

The performance of our proposed approach is measured using three metrics: precision \((TP / TP + FP)\), recall \((TP / TP + FN)\), and F1-value \((2 \times precision \times recall / precision + recall)\).
reviewer labels (the downstream task) to generate the final embeddings. The final hidden state of [CLS] (the classification token or class label) is the representation of the reviewer. To prepare the data, we tokenized each sentence and we removed the sentences with more than 256 words. As shown in Figs. 6 and 7, the handcrafted features can significantly affect the performance since these features fail to capture a comprehensive representation of groups. Such features mostly rely on simple statistics of a group such as group size or group rating deviation. The semantics of the written reviews by reviewers are overlooked, despite its proven effectiveness [11], [17]. We not only use the semantics in the reviewer-level representation but also a group-level representation of each group after excluding the deviant reviewers from the groups. Employing SoWE to represent reviewers results in capturing a global representation of reviewers, and consequently improves the fraudster group detection. Additionally, the group-level activities are better captured through advanced techniques such as SoWE or BERT, resulting in a more descriptive representation of groups. The exceptional performance of BERT is due to using a document-level corpus instead of shuffled sentences, resulting in a long contiguous dependency extraction, crucial in fraud detection.

3) HIN-RNN Performance in Graph Generation: To evaluate the performance of the HIN-RNN, we defined accuracy as a ratio of the number of correctly predicted edges $|E_c|$ over the total number of edges $|E|$

$$Acc. = \frac{|E_c|}{|E|}.$$  

(12)

We use accuracy to measure the performance of the HIN-RNN for the graph generation task. Fig. 8(a)–(d) show the accuracy on the node and edge prediction compared to the number of training epochs on the test set for both the Yelp and Amazon datasets. The plots demonstrate the effectiveness of the HIN-RNN (blue) in predicting the nodes and the edges compared to the GraphRNN (red). The results show that including the node type in the prediction through each node’s (reviewer representation in the current study) representation improves the node prediction by an average of 8% [Fig. 8(a)] on the Yelp dataset and 12% [Fig. 8(b)] on the Amazon dataset. In other words, including the representation of a reviewer considers the overlooked deep semantic contribution to the collaboration between the reviewers and hence improves the model to better predict reviewers in a group.

The edge prediction accuracy is also improved over the GraphRNN with an average of 15% on the Yelp dataset and an average of 8% on the Amazon dataset. Training the edges with nodes’ representation enables the HIN-RNN to rely on a corresponding feature space (acquired from the reviewer representation) to predict the possible collaboration. So two reviewers with similar representations and belonging to the same type have the same collaboration matrix, while two reviewers with different representations (and hence belong to different types) are unlikely to collaborate, even if two reviewers share the same connections as the candidate groups. The same trend also explains the fluctuation in training epochs.

We also conducted experimental evaluations on the number of correct nodes and the number of correct edges in groups with different sizes to show the effectiveness of the HIN-RNN in graph generation with respect to the graph complexity (size, number of edges) of the group. Fig. 9 shows the performance of the HIN-RNN on edge and node prediction against the group size and the number of edges for the Yelp and Amazon datasets. As Fig. 9(a) shows the HIN-RNN correctly predicts the nodes for smaller groups (up to four nodes) on the Yelp dataset, but as the number of nodes in a group increases the number of correctly predicted nodes decreases. On the other hand, the HIN-RNN correctly predicts all the nodes for different sizes on the Amazon dataset as shown in Fig. 9(b). We also displayed the performance of edge prediction in Fig. 9. Fig. 9(c) shows the performance of the HIN-RNN on the Yelp dataset for groups of different edge numbers. The HIN-RNN performance stays stable across groups with a different number of edges. Hence, we can claim that the HIN-RNN is capable of capturing the long range dependencies between the nodes in a group. As the number of edges increases, the HIN-RNN can still reliably model the relationships. Although the number of predicted edges on groups with a higher number of edges decreases with the Amazon dataset, the HIN-RNN is still capable of capturing the relationships between the nodes in a group at a higher number of edges, e.g., 22 edges. Adding the co-occurring semantic dependency between reviewers as a preknowledge results in a better relation modeling between different types of nodes in datasets, thus significantly improving the performance in predicting the nodes and the edges between reviewers.
4) Effects of Reviewers Removal: As explained, genuine reviewers are in some cases involved in unwanted fraudster group activities, and on the other hand, fraudsters may also camouflage themselves in genuine groups to escape detection. Figs. 10 and 11 show the distribution of the groups with only one fraudster or one genuine reviewer against the group size. As Fig. 10 shows, the number of groups with only one fraudster is much higher than groups with only one genuine reviewer. Additionally, single fraudsters are camouflaged in different groups regardless of the group size. This is intuitively possible since single fraudsters employ camouflage in any possible situation to escape detection. Genuine reviewers, on the other hand, contribute less to fraudster group activity and they mainly co-review with two fraudsters in a group with a size of three reviewers. But as Fig. 10 shows, genuine reviewers’ involvement is significantly decreased, as the group size grows. The Amazon dataset shows a similar trend in Fig. 11. Genuine reviewers are similarly involved in a fraudster activity for a group with sizes of 3–5, but fraudsters are involved in genuine groups across a wider range of group sizes. The single genuine reviewer in a fraudster group decreases the fraudster score of the group, which in turn increases FN-labeled samples. On the other hand, with fraudsters camouflaged in a genuine group, the fraudster score of a genuine group increases. This leads to an increase in FP-labeled samples.

With such an observation, in the final step (Section III-E), we first removed these deviant reviewers with minimum connections. In other words, we removed the reviewers with the least similarity to other reviewers in a group. We also conducted an experiment using a graph clustering method proposed by Wang et al. [40]. Wang et al. proposed an attention-based approach to capture the importance of each neighbor based on a loss function, and then the similarity of each node and the average of nodes’ representation in a group was calculated through another loss function. These loss functions are used to jointly train the framework to learn the clusters in an iterative algorithm. So we utilized the approach on each group to cluster each group into two clusters and then removed the cluster with minimum reviewers, as deviant reviewers. Figs. 12 and 13 show the performance of the proposed approach before and after deviant reviewer removal. The results suggest that the graph clustering has improved the performance of the proposed approach compared to the situation where reviewers are not removed.
Fig. 9. HIN-RNN performance on edge and node prediction versus group size and number of edges in each group. (a) Node prediction accuracy versus group size on Yelp. (b) Node prediction accuracy versus group size on Amazon. (c) Edge prediction accuracy versus group size on Yelp. (d) Edge prediction accuracy versus group size on Amazon.

Fig. 10. Distribution of groups with only one fraudster or only one genuine reviewer versus the group size on the Yelp dataset.

However, clustering does not perform as well as our proposed reviewer removal through minimum connections. Intuitively, the graph clustering shows the best performance on the large graphs and here most groups consist of 3–4 reviewers [see Fig. 9(a) and (b)]. This mostly results in a situation where no reviewer is removed from a group, because of the similar representation to each other. With more reviewers, more information on reviewers’ interaction is achieved, which

Fig. 11. Distribution of groups with only one fraudster or only one genuine reviewer vs. the group size on the Amazon dataset.
results in a better relation modeling of nodes in a group, thus better performance. As a result of removing unintentional genuine reviewers from fraudster groups and also fraudster imposters from a group of genuine reviewers, the FP and FN decrease, respectively. Hence, the precision and recall improve which in turn results in an improvement for F1-value for both Yelp and Amazon datasets.

V. CONCLUSION

Previous studies have investigated the problem of fraudster group detection, but they only rely on handcrafted features for groups, and they are not able to model the nonlocal semantic dependencies between reviewers in each group. In this study, we propose a four-step approach to address the challenges of this problem and improve performance: extracting reviewer representation, initializing candidate groups, collaboration modeling using an HIN-RNN, and finally removing the deviant reviewers from each group for final classification. The proposed approach outperforms most recent works by 22%, and 12% in terms of recall, and F1 on the Yelp dataset, respectively. Future works can use more advanced graph-based networks such as GCNs to refine the representations of reviewers in each group for final group-level representation. Another future direction could be a new approach to incorporate the representation for better collaboration matrix generation, instead of a simple concatenation.
Saeedreza Shehnepoor received the B.S. degree in information technology (IT) from the Sharif University of Technology, Tehran, Iran, in 2014, and the M.S. degree in computer science and information networks from the University of Tehran (UT), Tehran, in 2016. He is currently pursuing the Ph.D. degree with The University of Western Australia (UWA), Perth, WA, Australia.

His research interests include machine learning and specifically issues related to the fraud contents in social media using graph theory and deep learning.

Wei Liu (Member, IEEE) received the Ph.D. degree from The University of Newcastle, Callaghan NSW, Australia, in 2003.

She is currently a full-time Teaching and Research Academic with the Department of Computer Science and Software Engineering, The University of Western Australia, Perth, WA, Australia. She has won three Australian Research Council (ARC) Grants and managed several industry grants. Her research focuses on knowledge discovery from natural language text, deep learning methods for knowledge graph construction and analysis, and sequential data mining and forecasting in traffic and water consumption domain. Her current industry-related research projects include knowledge graph refinement for geological survey reports, incident log analysis and visualization, short-term traffic prediction, and cognitive computing for asset management.

Mohammed Bennamoun (Senior Member, IEEE) received the M.Sc. degree in control theory from Queen’s University, Kingston, ON, Canada, in 1992, and the Ph.D. degree in computer vision from the Queen’s University of Technology (QUT), Brisbane, QLD, Australia, in 1996.

In 1993, he lectured robotics at Queen’s University, and then joined QUT, as an Associate Lecturer. From February 2007 to March 2012, he was the Head of the School of Computer Science and Software Engineering, The University of Western Australia (UWA), Perth, WA, Australia. He is currently a Winthrop Professor. His areas of interest include computer vision (particularly 3-D), such as object recognition and biometrics, machine/deep learning, and robotics, such as obstacle avoidance and robot grasping, signal/image processing, and control theory.

Dr. Bennamoun served as a member for the Australian Research Council (ARC) College of Experts from 2012 to 2015 and from 2019 to 2021. He also served as a member for the ARC Excellence Research for Australia (ERA) in 2018. He is also a member of the College of Assessors for the Ministry of Business, Innovation and Employment, New Zealand.

Roberto Togneri (Senior Member, IEEE) received the Ph.D. degree from The University of Western Australia, Perth, WA, Australia, in 1989.

In 1988, he joined the School of Electrical, Electronic and Computer Engineering, The University of Western Australia, where he is currently an Associate Professor. He also leads the Signal Processing and Recognition Laboratory, Department of Electrical, Electronics and Computer Engineering, The University of Western Australia. He is the Chief Investigator on three Australian Research Council (AUC) Discovery Project Research Grants. His research activities in signal processing and pattern recognition include feature learning, detection and enhancement of audio signals, statistical and neural network models for speech and speaker recognition, audio-visual recognition and biometrics, and related aspects of language modeling and understanding.

Dr. Togneri is the Area Editor of IEEE Signal Processing Magazine Columns and Forums.