Node Classification in Complex Social Graphs via Knowledge-Graph Embeddings and Convolutional Neural Network

Bonaventure C. Molokwu[1(✉)], Shaon Bhatta Shuvo[1], Narayan C. Kar[2], and Ziad Kobti[1]

[1] School of Computer Science, University of Windsor, 401 Sunset Avenue, Windsor, ON N9B-3P4, Canada
{molokwub,shuvos,kobti}@uwindsor.ca
[2] Centre for Hybrid Automotive Research and Green Energy (CHARGE), University of Windsor, 401 Sunset Avenue, Windsor, ON N9B-3P4, Canada
nkar@uwindsor.ca

Abstract. The interactions between humans and their environment, comprising living and non-living entities, can be studied via Social Network Analysis (SNA). Node classification, as well as community detection tasks, are still open research problems in SNA. Hence, SNA has become an interesting and appealing domain in Artificial Intelligence (AI) research. Immanent facts about social network structures can be effectively harnessed for training AI models in a bid to solve node classification and community detection problems in SNA. Hence, crucial aspects such as the individual attributes of spatial social actors, and the underlying patterns of relationship binding these social actors must be taken into consideration in the course of analyzing the social network. These factors determine the nature and dynamics of a given social network. In this paper, we have proposed a unique framework, Representation Learning via Knowledge-Graph Embeddings and ConvNet (RLVECN), for studying and extracting meaningful facts from social network structures to aid in node classification as well as community detection tasks. Our proposition utilizes an edge sampling approach for exploiting features of the social graph, via learning the context of each actor with respect to neighboring actors/nodes, with the goal of generating vector-space embedding per actor. Successively, these relatively low-dimensional vector embeddings are fed as input features to a downstream classifier for classification tasks about the social graph/network. Herein RLVECN has been trained, tested, and evaluated on real-world social networks.

Keywords: Node classification · Feature learning · Feature extraction · Dimensionality reduction · Semi-supervised learning

This research was supported by International Business Machines (IBM) and Compute Canada (SHARCNET).

V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12142, pp. 183–198, 2020.
https://doi.org/10.1007/978-3-030-50433-5_15

1 Introduction and Related Literature

Humans inhabit in a planet comprised of several systems and ecosystems; and interaction is a natural phenomenon and characteristic obtainable in any given system or ecosystem. Thus, relationships between constituent entities in a given system/ecosystem is a strategy for survival and essentiality for the sustenance of a given system/ecosystem. Owing to the recent AI advances, these real-world complex systems and ecosystems can be effectively modelled as social network structures for analysis. Social network graphs [22] are intricate structures which present analytical challenges to Machine Learning (ML) and Deep Learning (DL) models because of their dynamic nature, complex links, and occasionally massive size. In this regard, we have proposed a new hybrid and DL-based model (RLVECN) based on an iterative learning approach [1] for solving (node) classification as well as clustering problems in SNA via an edge sampling strategy.

In SNA, the classification of nodes induces the formation of cluster(s). Consequently, clusters give rise to homophily in social networks. Basically, learning in RLVECN is induced via semi-supervised training. The architecture of RLVECN comprises two (2) distinct Representation Learning (RL) layers, viz: a Knowledge-Graph Embeddings (VE) layer and a Convolutional Neural Network (ConvNet) layer [13]. Both of these layers are trained by means of unsupervised learning. These layers are essentially feature-extraction and dimensionality-reduction layers where underlying knowledge and viable facts are automatically extracted from the social network structure [15]. The vector-embedding layer is responsible for projecting the feature representation of the social graph to a q-dimensional real-number space, $\mathbb{R}^q$. This is done by associating a real-number vector to every unique actor/node in the social network such that the cosine distance of any given tie/edge (a pair of actors) would capture a significant degree of correlation between the two associated actors. Furthermore, the ConvNet layer, which feeds on the vector-embedding layer, is responsible for further extraction of apparent features and/or representations from the social graph. Finally, a classification layer succeeds the RL layers; and it is trained by means of supervised learning. The classifier is based on a Neural Network (NN) architecture assembled using multiple and deep layers of stacked perceptrons (NN units) [6]. Every low-dimensional feature (X), extracted by the representation-learning layers, is mapped to a corresponding output label (Y); and these (X, Y) pairs are used to supervise the training of the classifier such that it can effectively and efficiently learn how to identify clusters and classify actors within a given social network structure. Hence, the novelty of our research contribution are as stated below:

(1) Proposition of a DL-based and hybrid model, RLVECN, which is aimed at solving node classification problems in SNA.
(2) Detailed benchmarking reports with respect to classic objective functions used for classification tasks.
(3) Comparative analysis, between RLVECN and state-of-the-art methodologies, against standard real-world social networks.

RLVECN is capable of learning the non-linear distributed features enmeshed in a social network [9]. We have evaluated RLVECN against an array of state-of-the-art models and RL methodologies which serve as our baselines, viz:

(i) DeepWalk: Online Learning of Social Representations [19].
(ii) GCN: Semi-Supervised Classification with Graph Convolutional Networks [11].
(iii) LINE: Large-scale Information Network Embedding [25].
(iv) Node2Vec: Scalable Feature Learning for Networks [8].
(v) SDNE: Structural Deep Network Embedding [26].

2 Proposed Methodology and Framework

2.1 Definition of Problem

Definition 1. *Social Network, SN: As expressed via Eq. 1 such that SN is a tuple comprising a set of actors/vertices, V; a set of ties/edges, E; a metadata function, f_V, which extends the definition of the vertices' set by mapping it to a given set of attributes, V′; and a metadata function, f_E, which extends the definition of the edges' set by mapping it to a given set of attributes, E′. Thus, a graph function, $G(V, E) \subset SN$*

$$\begin{aligned} SN &= (V, E, f_V, f_E) \equiv (G, f_V, f_E) \\ V &: |\{V\}| = M && \text{set of actors/vertices with size, M} \\ E &: E \subset \{U \times V\} \subset \{V \times V\} && \text{set of ties/edges between V} \\ f_V &: V \rightarrow V' && \text{vertices' metadata function} \\ f_E &: E \rightarrow E' && \text{edges' metadata function} \end{aligned} \tag{1}$$

Definition 2. *Knowledge Graph, KG: $(\mathbb{E}, \mathbb{R})$ is a set comprising entities, $\mathbb{E}$, and relations, $\mathbb{R}$, between the entities. Thus, a KG [24,28] is defined via a set of triples, $t : (u, p, v)$, where $u, v \in \mathbb{E}$ and $p \in \mathbb{R}$. Also, a KG [27] can be modelled as a social network, SN, such that: $\mathbb{E} \rightarrow V$ and $\mathbb{R} \rightarrow E$ and $(\mathbb{E}, \mathbb{R}) \vdash f_V, f_E$.*

Definition 3. *Knowledge-Graph (Vector) Embeddings, X: The vector-space embeddings, X, generated by the embedding layer are based on a mapping function, f, expressed via Eq. 2. f projects the representation of the graph's actors to a q-dimensional real space, $\mathbb{R}^q$, such that the existent ties between any given pair of actors, (u_i, v_j), remain preserved via the homomorphism from V to X.*

$$\begin{aligned} &f : V \rightarrow X \in \mathbb{R}^q \\ &f : (u, p, v) \rightarrow X \in \mathbb{R}^q \quad \text{Knowledge-Graph Embeddings} \end{aligned} \tag{2}$$

Definition 4. *Node Classification: Considering, SN, comprising partially labelled actors (or vertices), $V_{lbl} \subset V : V_{lbl} \rightarrow Y_{lbl}$; and unlabelled vertices defined such that: $V_{ulb} = V - V_{lbl}$. A node-classification task aims at training a predictive function, $f : V \rightarrow Y$, that learns to predict the labels, Y, for all actors or vertices, $V \subset SN$, via knowledge harnessed from the mapping: $V_{lbl} \rightarrow Y_{lbl}$.*

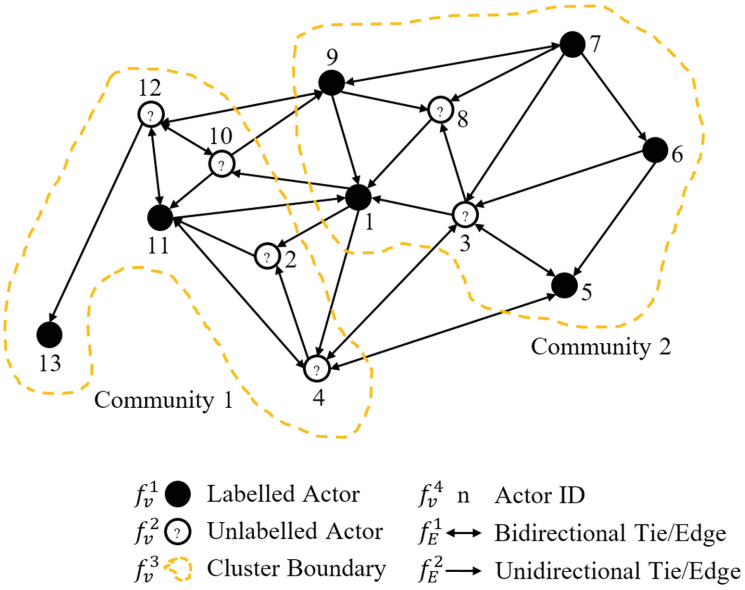

Fig. 1. Node classification task in social graphs

2.2 Proposed Methodology

Our proposition, RLVECN, is comprised of two (2) distinct Feature Learning (FL) layers, and one (1) classification layer (Fig. 2).

Representation Learning - Knowledge-Graph Embeddings Layer: Given a social network, SN, defined by a set of actors/vertices, $V : U \subset V \forall \{u_m, v_m\} \in V$, and $M : m \in M$ denotes the number of unique actors in SN. Additionally, let the ties/edges in SN be defined such that: $E \subset \{U \times V\}$; where $u_i \in V$ and $v_j \in V$ represent a source_vertex and a target_vertex in E, respectively.

The objective function of the vector-embedding layer aims at maximizing the average logarithmic probability of the source_vertex, u_i, being predicted as neighboring actor to the target_vertex, v_j, with respect to all training pairs, $\forall (u_i, v_j) \in E$. Formally, the function is expressed as in Eq. 3:

$$\mu = \frac{1}{M} \sum_{m=1}^{M} (\sum_{(u_i, v_j) \in E} logPr(u_i|v_j)) \tag{3}$$

Consequently, in order to compute $Pr(u_i|v_j)$, we have to quantify the proximity of each target_vertex, v_j, with respect to its source_vertex, u_i. The vector-embedding model measures this adjacency/proximity as the cosine similarity between v_j and its corresponding u_i. Thus, the cosine distance is calculated as

the dot product between the target_vertex and the source_vertex. Mathematically, $Pr(u_i|v_j)$ is computed via a softmax function as defined in Eq. 4:

$$Pr(u_i|v_j) = \frac{exp(u_i \cdot v_j)}{\sum_{m=1}^{M} exp(u_m \cdot v_j)} \tag{4}$$

Hence, the objective function of our vector-embedding model with respect to the SN is as expressed by Eq. 5:

$$\sum_{(u_i,v_j)\in E} logPr(u_i|v_j) = \sum_{(u_i,v_j)\in E} log\frac{exp(u_i \cdot v_j)}{\sum_{m=1}^{M} exp(u_m \cdot v_j)} \tag{5}$$

Representation Learning - ConvNet Layer: This layer comprises three (3) RL or FL operations, namely: convolution, non-linearity, and pooling operations. RLVECN utilizes a one-dimensional (1D) convolution layer [14] which is sandwiched between the vector-embedding and classification layers. Equation 6 expresses the 1D-convolution operation:

$$\begin{gathered} FeatureMap(F) = 1D\_InputMatrix(X) * Kernel(K) \\ f_i = (X * K)_i = (K * X)_i = \sum_{j=0}^{J-1} x_j \cdot k_{i-j} = \sum_{j=0}^{J-1} k_j \cdot x_{i-j} \end{gathered} \tag{6}$$

where f_i represents a cell/matrix position in the Feature Map; k_j denotes a cell position in the Kernel; and x_{i-j} denotes a cell/matrix position in the 1D-Input (data) matrix.

The non-linearity operation is a rectified linear unit (ReLU) function which introduces non-linearity after the convolution operation since real-world problems usually exist in non-linear form(s). As a result, the rectified feature/activation map is computed via: $r_i \in R = g(f_i \in F) = max(0, F)$.

The pooling operation is responsible for reducing the input width of each rectified activation map while retaining its vital properties. In this regard, the *Max Pooling* function is defined such that the resultant pooled (or downsampled) feature map is generated via: $p_i \in P = h(r_i \in R) = maxPool(R)$.

Classification - Multi-Layer Perceptron (MLP) Classifier Layer: This is the last layer of our proposed RLVECN's architecture, and it succeeds the RL layers. The pooled feature maps, generated by the Representation Learning layers, contain high-level features extracted from the constituent actors of the social network structure. Hence, the classification layer utilizes these extracted "high-level features" for identifying clusters, based on the respective classes, contained in the social graph. In this regard, a MLP [5] function is defined as a mathematical function, f_c, mapping some set of input values, P, to their respective output labels, Y. In other words, $Y = f_c(P, \Theta)$, and Θ denotes a set of parameters. The MLP [4] function learns the values of Θ that will result in the

best decision, Y, approximation for the input set, P. The MLP classifier output is a probability distribution which indicates the likelihood of a feature representation belonging to a particular output class. Our MLP [10] classifier is modelled such that sequential layers of NN units are stacked against each other to form a Deep Neural Network (DNN) structure [3,16].

Node Classification Algorithm: Defined via Algorithm 1.

Algorithm 1. Proposed Algorithm for Node Classification

Input: $\{V, E, Y_{lbl}\} \equiv$ {Actors, Ties, Ground-Truth Labels}
Output: $\{Y_{ulb}\} \equiv$ {Predicted Labels}

Preprocessing:
$V_{lbl}, V_{ulb} \subset V = V_{lbl} \cup V_{ulb}$ // V_{lbl} : Labelled actors // V_{ulb} : Unlabelled actors

$E : (u_i, v_j) \in \{U \times V\}$ // $(u_i, v_j) \equiv$ (source, target)
$E_{train} = E_t : u_i, v_j \in V_{lbl}$ // $|E_{train}| = \sum indegree(V_{lbl}) + \sum outdegree(V_{lbl})$
$E_{pred} = E_p : u_i, v_j \in V_{ulb}$

$f_c \leftarrow$ Initialize // Construct classifier model

Training:
for $t \leftarrow 0$ **to** $|E_{train}|$ **do**
  $f : E_t \rightarrow [X \in \mathbb{R}^q]$ // Embedding operation
  $f_t \in F = (K * X)_t$ // Convolution operation
  $r_t \in R = g(F) = max(0, f_t)$
  $p_t \in P = h(R) = maxPool(r_t)$
  $f_c|\Theta : p_t \rightarrow Y_{lbl}$ // MLP classification operation
end for

return $Y_{ulb} = f_c(E_{pred}, \Theta)$

2.3 Proposed Architecture/Framework

Figure 2 illustrates the architecture of our proposition, RLVECN.

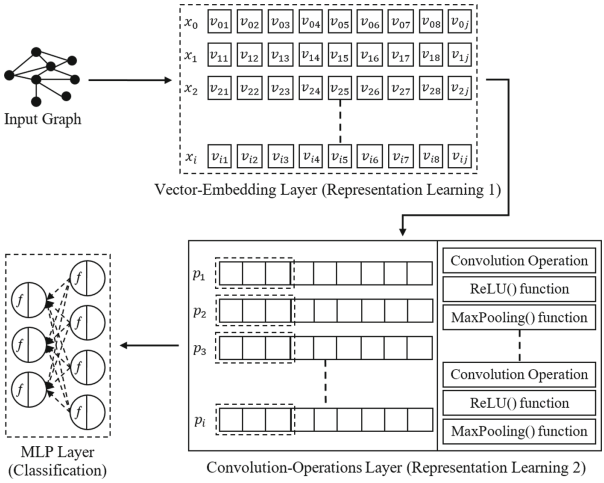

Fig. 2. Proposed system architecture

3 Data Sets and Materials

3.1 Data Sets

With regard to Table 1 herein, six (6) real-world benchmark social-graph data sets were employed for experimentation and evaluation, viz: Cora [20,23], CiteSeer [20,23], Facebook Page-Page webgraph [21], PubMed-Diabetes [17], Internet-Industry partnerships [2,12], and Terrorists-Relationship [29].

Table 1. Benchmark data sets

Data set	Classes → {label: 'description'}
Cora	$G(V, E) = G(2708, 5429)$
	{C1: 'Case_Based', C2: 'Genetic_Algorithms', C3: 'Neural_Networks', C4: 'Probabilistic_Methods', C5: 'Reinforcement_Learning', C6: 'Rule_Learning', C7: 'Theory'}
CiteSeer	$G(V, E) = G(3312, 4732)$
	{C1: 'Agents', C2: 'Artificial Intelligence', C3: 'Databases', C4: 'Information Retrieval', C5: 'Machine Learning', C6: 'Human-Computer Interaction'}
Facebook Page2Page	$G(V, E) = G(22470, 171002)$
	{C1: 'Companies', C2: 'Governmental Organizations', C3: 'Politicians', C4: 'Television Shows'}
PubMed Diabetes	$G(V, E) = G(19717, 44338)$
	{C1: 'Diabetes Mellitus - Experimental', C2: 'Diabetes Mellitus - Type 1', C3: 'Diabetes Mellitus - Type 2'}
Internet Industry	$G(V, E) = G(219, 631)$
	{C1: 'Content Sector', C2: 'Infrastructure Sector', C3: 'Commerce Sector'}
Terrorists Relation	$G(V, E) = G(851, 8592)$
	{C1: 'Content Sector', C2: 'Infrastructure Sector', C3: 'Commerce Sector'}

3.2 Data Preprocessing

All benchmark data sets ought to be comprised of actors and ties already encoded as discrete data (natural-number format). However, Cora, CiteSeer, Facebook-Page2Page, PubMed-Diabetes, and Terrorists-Relation data sets are made up of nodes and/or edges encoded in mixed formats (categorical and numerical formats). Thus, it is necessary to transcode these non-numeric (categorical) entities to their respective discrete (numeric) data representation, without semantic loss, via an injective function that maps each distinct entry in the categorical-entity domain to distinct numeric values in the discrete-data codomain, $f_m : categorical \rightarrow discrete$. Thereafter, the numeric representation of all benchmark data sets are normalized, $f_n : discrete \rightarrow continuous$, prior to training against RLVECN and the benchmark models. Also, only edgelist ties, $E \subset G$, whose constituent actors are present in the nodelist, $V \subset G$, were used for training/testing our model.

Table 2. Configuration of experimentation hyperparameters

Training Set: 80%	Test Set: 20%	Network Width: 640
Batch Size: 256	Optimizer: $AdaMax$	Network Depth: 6
Epochs: $1.8 * 10^2$	Activation: $ReLU$	Dropout: $4.0 * 10^{-1}$
Learning Rate: $1.0 * 10^{-3}$	Learning Decay: 0.0	Embed Dimension: 100

4 Experiment, Results, and Discussions

RLVECN's experimentation setup was tuned in accordance with the hyperparameters shown in Table 2. Our evaluations herein were recorded with reference to a range of objective functions. Thus, Categorical Cross Entropy was employed as the cost/loss function; while the fitness/utility was measured based on the following metrics: Precision (PC), Recall (RC), F-measure or F1-score (F1), Accuracy (AC), and Area Under the Receiver Operating Characteristic Curve (RO). Moreover, the objective functions have been computed against each benchmark data set with regard to the constituent classes (or categories) present in each data set. The Support (SP) represents the number of ground-truth samples per class/category for each data set.

In a bid to avoid sample bias across-the-board, we have used exactly the same SP for all models inclusive of RLVECN model. However, since RLVECN is based on an edge-sampling technique; the SP recorded against RLVECN model represent the numbers of edges/ties used for computation as explained in Algorithm 1. With regard to the standard node-classification tasks herein, the performance of our RLVECN model during benchmarking against five(5) popular baselines (DeepWalk, GCN, LINE, Node2Vec, SDNE); and when evaluated against the validation/test samples for the benchmark data sets are as documented in Table 3, 4, 5, and 6 respectively. Consequently, Fig. 3 graphically shows the

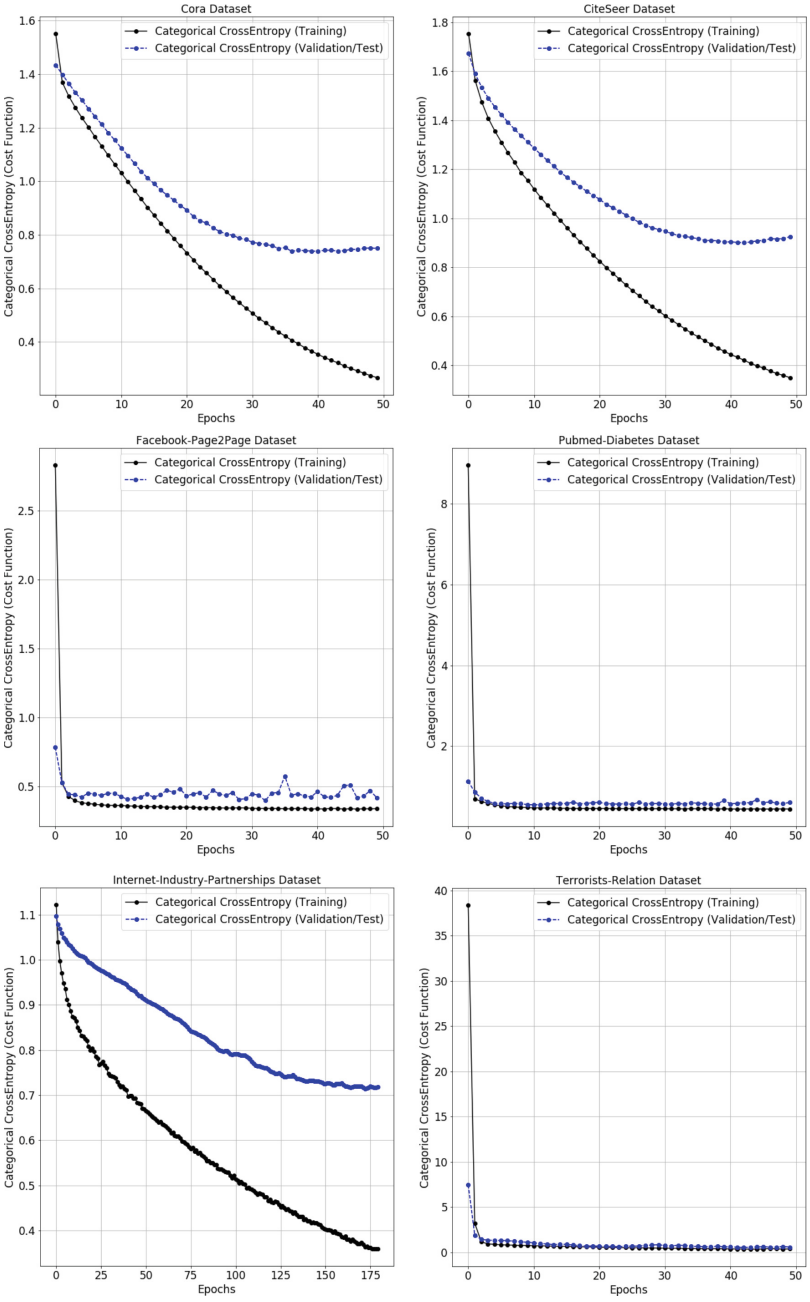

Fig. 3. RLVECN's learning curves during node-classification training against Cora, CiteSeer, Facebook-Page2Page, PubMed-Diabetes, Internet-Industry-Partnership, and Terrorists-Relation data sets (loss function *vs* training epochs)

Table 3. Node-classification over Cora data set. Results are based on the set apart validation sample - data set *vs* models.

Model	Metric	Cora Dataset								Points
		C1	C2	C3	C4	C5	C6	C7	μ	
RLVECN	PC	0.85	0.78	0.80	0.88	0.72	0.90	0.81	0.82	14
	RC	0.86	0.93	0.81	0.87	0.75	0.91	0.78	0.84	
	F1	**0.86**	**0.85**	**0.81**	**0.87**	**0.74**	**0.91**	**0.79**	0.83	
	AC	0.93	0.98	0.96	0.95	0.93	0.97	0.96	0.95	
	RO	**0.90**	**0.96**	**0.90**	**0.92**	**0.85**	**0.95**	**0.88**	0.91	
	SP	541	134	214	405	294	345	237	310	
GCN	PC	0.87	0.95	0.89	0.92	0.85	0.89	0.87	0.89	3
	RC	0.85	0.73	0.65	0.82	0.58	0.85	0.73	0.74	
	F1	**0.86**	0.83	0.75	**0.87**	0.69	0.87	**0.79**	0.81	
	AC	0.89	0.93	0.91	0.92	0.88	0.93	0.91	0.91	
	RO	0.88	0.83	0.80	0.89	0.75	0.90	0.83	0.84	
	SP	164	36	43	85	70	84	60	77	
Node2Vec	PC	0.58	0.78	0.72	0.81	0.80	0.84	0.82	0.76	0
	RC	0.85	0.50	0.53	0.68	0.64	0.74	0.60	0.65	
	F1	0.69	0.61	0.61	0.74	0.71	0.78	0.69	0.69	
	AC	0.77	0.96	0.95	0.92	0.93	0.94	0.94	0.92	
	RO	0.79	0.75	0.76	0.83	0.81	0.86	0.79	0.80	
	SP	164	36	43	85	70	84	60	77	
DeepWalk	PC	0.57	0.58	0.72	0.58	0.68	0.72	0.63	0.64	0
	RC	0.80	0.42	0.42	0.59	0.39	0.63	0.65	0.56	
	F1	0.67	0.48	0.53	0.58	0.49	0.67	0.64	0.58	
	AC	0.76	0.94	0.94	0.87	0.90	0.90	0.92	0.89	
	RO	0.77	0.70	0.70	0.75	0.68	0.79	0.80	0.74	
	SP	164	36	43	85	70	84	60	77	
LINE	PC	0.35	0.86	0.80	0.65	0.50	0.43	0.61	0.60	0
	RC	0.85	0.17	0.19	0.35	0.20	0.15	0.23	0.31	
	F1	0.50	0.28	0.30	0.46	0.29	0.23	0.34	0.34	
	AC	0.48	0.94	0.93	0.87	0.87	0.84	0.90	0.83	
	RO	0.59	0.58	0.59	0.66	0.59	0.56	0.61	0.60	
	SP	164	36	43	85	70	84	60	77	
SDNE	PC	0.37	0.83	0.70	0.60	0.54	0.64	0.64	0.62	0
	RC	0.91	0.14	0.16	0.35	0.20	0.27	0.12	0.31	
	F1	0.53	0.24	0.26	0.44	0.29	0.38	0.20	0.33	
	AC	0.50	0.94	0.93	0.86	0.87	0.86	0.89	0.84	
	RO	0.62	0.57	0.58	0.65	0.59	0.62	0.55	0.60	
	SP	164	36	43	85	70	84	60	77	

Table 4. Node-classification over CiteSeer data set. Results are based on the set apart test sample - data set *vs* models.

Model	Metric	CiteSeer Dataset							Points
		C1	C2	C3	C4	C5	C6	μ	
RLVECN	PC	0.76	0.81	0.78	0.43	0.88	0.60	0.71	12
	RC	0.84	0.83	0.79	0.60	0.79	0.65	0.75	
	F1	**0.80**	**0.82**	**0.79**	**0.50**	**0.83**	**0.63**	0.73	
	AC	0.93	0.88	0.92	0.93	0.96	0.89	0.92	
	RO	**0.90**	**0.87**	**0.87**	**0.78**	**0.89**	**0.79**	0.85	
	SP	304	609	377	107	225	275	316	
GCN	PC	0.80	0.78	0.86	0.95	0.91	0.75	0.84	2
	RC	0.76	0.76	0.73	0.08	0.67	0.54	0.59	
	F1	0.78	0.77	**0.79**	0.15	0.77	**0.63**	0.65	
	AC	0.88	0.87	0.88	0.91	0.89	0.83	0.88	
	RO	0.84	0.83	0.83	0.53	0.81	0.72	0.76	
	SP	119	134	140	50	102	118	111	
Node2Vec	PC	0.57	0.55	0.49	0.33	0.55	0.38	0.48	0
	RC	0.55	0.60	0.66	0.06	0.45	0.40	0.45	
	F1	0.56	0.58	0.56	0.10	0.50	0.39	0.45	
	AC	0.85	0.82	0.78	0.92	0.86	0.78	0.84	
	RO	0.73	0.74	0.74	0.53	0.69	0.63	0.68	
	SP	119	134	140	50	102	118	111	
DeepWalk	PC	0.46	0.53	0.43	0.43	0.47	0.33	0.44	0
	RC	0.51	0.54	0.57	0.06	0.41	0.32	0.40	
	F1	0.49	0.54	0.49	0.11	0.44	0.32	0.40	
	AC	0.81	0.81	0.75	0.92	0.84	0.76	0.82	
	RO	0.69	0.71	0.69	0.53	0.66	0.59	0.65	
	SP	119	134	140	50	102	118	111	
SDNE	PC	0.37	0.50	0.24	0.20	0.45	0.31	0.35	0
	RC	0.19	0.27	0.77	0.02	0.14	0.09	0.25	
	F1	0.25	0.35	0.36	0.04	0.21	0.14	0.23	
	AC	0.80	0.80	0.42	0.92	0.84	0.80	0.76	
	RO	0.56	0.60	0.55	0.51	0.55	0.52	0.55	
	SP	119	134	140	50	102	118	111	
LINE	PC	0.18	0.30	0.28	0.60	0.22	0.27	0.31	0
	RC	0.15	0.47	0.39	0.06	0.12	0.21	0.23	
	F1	0.16	0.36	0.32	0.11	0.15	0.24	0.22	
	AC	0.72	0.67	0.65	0.93	0.80	0.76	0.76	
	RO	0.50	0.59	0.56	0.53	0.52	0.55	0.54	
	SP	119	134	140	50	102	118	111	

learning-progress curves during the node-classification tasks using our proposed RLVECN model; and when training over the benchmark data sets. Hence, the dotted-black lines represent learning progress over the training set; and the dotted-blue lines represent learning progress over the test set.

Tables 3, 4, 5, and 6 have clearly tabulated our results as a multi-classification task over the benchmark data sets. Thus, for each class per data set, we have laid emphasis on the F1 (the weighted average of the PC and RC metrics) and

Table 5. Node-classification experiment over Facebook Page-Page webgraph and PubMed-Diabetes data sets. Results are based on the reserved test sample - data set *vs* models. N.B.: Mtc = Metric (fitness function); Pts = Points.

Model	Mtc	Facebook-Page2Page					Pts	PubMed-Diabetes				Pts
		C1	C2	C3	C4	μ		C1	C2	C3	μ	
RLVECN	PC	0.87	0.95	0.91	0.87	0.90	8	0.76	0.83	0.84	0.81	6
	RC	0.84	0.85	0.85	0.86	0.85		0.60	0.88	0.91	0.80	
	F1	**0.85**	**0.90**	**0.88**	**0.86**	0.87		**0.67**	**0.86**	**0.87**	0.80	
	AC	0.96	0.90	0.94	0.97	0.94		0.89	0.88	0.90	0.89	
	RO	**0.97**	**0.97**	**0.98**	**0.98**	0.98		**0.92**	**0.94**	**0.95**	0.94	
	SP	9989	33962	16214	6609	16694		3300	7715	7170	6062	
Node2Vec	PC	0.81	0.84	0.81	0.84	0.83	0	0.74	0.47	0.49	0.57	0
	RC	0.82	0.87	0.85	0.67	0.80		0.03	0.65	0.55	0.41	
	F1	0.81	0.85	0.83	0.74	0.81		0.05	0.55	0.52	0.37	
	AC	0.89	0.91	0.91	0.93	0.91		0.80	0.57	0.60	0.66	
	RO	0.87	0.90	0.89	0.82	0.87		0.51	0.58	0.59	0.56	
	SP	1299	1376	1154	665	1124		821	1575	1548	1315	
DeepWalk	PC	0.75	0.84	0.76	0.75	0.78	0	0.65	0.57	0.58	0.60	0
	RC	0.81	0.85	0.82	0.52	0.75		0.15	0.67	0.71	0.51	
	F1	0.78	0.84	0.79	0.62	0.76		0.24	0.62	0.63	0.50	
	AC	0.87	0.90	0.89	0.90	0.89		0.81	0.67	0.68	0.72	
	RO	0.85	0.89	0.87	0.75	0.84		0.56	0.67	0.69	0.64	
	SP	1299	1376	1154	665	1124		821	1575	1548	1315	
LINE	PC	0.53	0.66	0.72	0.66	0.64	0	0.48	0.42	0.44	0.45	0
	RC	0.72	0.71	0.59	0.29	0.58		0.05	0.60	0.46	0.37	
	F1	0.61	0.68	0.65	0.40	0.59		0.08	0.50	0.45	0.34	
	AC	0.73	0.80	0.83	0.87	0.81		0.79	0.51	0.56	0.62	
	RO	0.73	0.77	0.75	0.63	0.72		0.52	0.53	0.54	0.53	
	SP	1299	1376	1154	665	1124		821	1575	1548	1315	
SDNE	PC	0.49	0.80	0.70	0.65	0.66	0	0.65	0.43	0.74	0.61	0
	RC	0.90	0.63	0.50	0.19	0.56		0.05	0.96	0.17	0.39	
	F1	0.64	0.70	0.58	0.29	0.55		0.10	0.59	0.27	0.32	
	AC	0.70	0.84	0.82	0.86	0.81		0.80	0.48	0.65	0.64	
	RO	0.76	0.78	0.71	0.58	0.71		0.52	0.56	0.56	0.55	
	SP	1299	1376	1154	665	1124		821	1575	1548	1315	

Table 6. Node-classification experiment over Internet-Industry-Partnership and Terrorists-Relationship data sets. Results are based on the reserved validation sample - data set *vs* models. N.B.: Mtc = Metric (fitness function); Pts = Points.

Model	Mtc	Internet-Industry-Partnership				Pts	Terrorists-Relation Dataset					Pts
		C1	C2	C3	μ		C1	C2	C3	C4	μ	
RLVECN	PC	0.33	0.96	0.29	0.53	5	0.93	0.91	0.46	1.00	0.83	6
	RC	0.65	0.77	0.76	0.73		0.97	0.97	0.42	0.97	0.83	
	F1	**0.44**	**0.86**	0.42	0.57		**0.95**	**0.94**	0.44	**0.98**	0.83	
	AC	0.84	0.78	0.87	0.83		0.95	0.98	0.89	0.99	0.95	
	RO	**0.76**	**0.81**	**0.82**	0.80		**0.98**	**1.00**	0.85	**1.00**	0.96	
	SP	26	238	17	94		1706	491	319	561	769	
GCN	PC	No experiment due to the absence of vectorized feature set for this data set				0	0.94	0.74	0.67	0.96	0.83	5
	RC						0.90	0.95	0.60	1.00	0.86	
	F1						0.92	0.83	**0.63**	**0.98**	0.84	
	AC						0.92	0.95	0.88	0.99	0.94	
	RO						**0.98**	0.99	**0.91**	**1.00**	0.97	
	SP						92	21	30	27	43	
DeepWalk	PC	0.50	0.81	0.36	0.56	1	0.88	0.82	0.64	0.86	0.80	0
	RC	0.12	0.93	0.44	0.50		0.90	0.86	0.53	0.93	0.81	
	F1	0.20	**0.86**	0.40	0.49		0.89	0.84	0.58	0.89	0.80	
	AC	0.82	0.82	0.73	0.79		0.88	0.96	0.86	0.96	0.92	
	RO	0.55	0.79	0.62	0.65		0.88	0.92	0.73	0.95	0.87	
	SP	8	27	9	15		92	21	30	27	43	
Node2Vec	PC	0.00	0.68	0.75	0.48	1	0.86	0.82	0.60	0.86	0.79	0
	RC	0.00	1.00	0.33	0.44		0.88	0.86	0.50	0.93	0.79	
	F1	0.00	0.81	**0.46**	0.42		0.87	0.84	0.55	0.89	0.79	
	AC	0.82	0.70	0.84	0.79		0.86	0.96	0.85	0.96	0.91	
	RO	0.50	0.62	0.65	0.59		0.86	0.92	0.71	0.95	0.86	
	SP	8	27	9	15		92	21	30	27	43	
LINE	PC	0.00	0.61	0.00	0.20	0	0.82	0.82	0.58	0.92	0.79	0
	RC	0.00	1.00	0.00	0.33		0.92	0.86	0.37	0.85	0.75	
	F1	0.00	0.76	0.00	0.25		0.87	0.84	0.45	0.88	0.76	
	AC	0.82	0.61	0.80	0.74		0.85	0.96	0.84	0.96	0.90	
	RO	0.50	0.50	0.50	0.50		0.84	0.92	0.65	0.92	0.83	
	SP	8	27	9	15		92	21	30	27	43	
SDNE	PC	0.00	0.61	0.00	0.20	0	0.77	0.90	0.56	1.00	0.81	0
	RC	0.00	1.00	0.00	0.33		0.92	0.86	0.30	0.85	0.73	
	F1	0.00	0.76	0.00	0.25		0.84	0.88	0.39	0.92	0.76	
	AC	0.82	0.61	0.80	0.74		0.81	0.97	0.84	0.98	0.90	
	RO	0.50	0.50	0.50	0.50		0.80	0.92	0.62	0.93	0.82	
	SP	8	27	9	15		92	21	30	27	43	

RO. Therefore, we have highlighted the model which performed best (based on F1 and RO metrics) for each classification task using a **bold font**. Additionally, we have employed a point-based ranking standard to ascertain the fittest model for each node classification task. The model with the highest aggregate point signifies the fittest model for the specified task, and so on in a descending order of aggregate points. Accordingly, as can be seen from our tabular results, our proposed methodology (RLVECN) is at the top with the highest fitness points.

$L2$ regularization ($L2 = 0.04$) [7] and early stopping [18] were used herein as addon regularization techniques to overcome overfitting incurred during RLVECN's training. Hence, the application of early stopping with respect to RLVECN's training over the benchmark data sets were, viz: 50 epochs (Cora, CiteSeer, Facebook-Page2Page, PubMed-Diabetes, Terrorists-Relation) and 180 epochs (Internet-Industry-Partnership). We have used a mini-batch size of 256 for training and testing/validating because we want to ensure that sufficient patterns are extracted by RLVECN during training before its network weights are updated.

5 Limitations, Conclusion, Future Work, and Acknowledgements

The benchmark models evaluated herein were implemented using their default parameters. We were not able to evaluate GCN [11] against Facebook Page-Page webgraph, PubMed-Diabetes, and Internet-Industry-Partnership data sets; because each of these aforementioned data sets does not possess a vectorized feature set which is required by GCN model for input-data processing. Overall, RLVECN's remarkable performance with respect to our benchmarking results can be attributed to the following:

(1) The RL kernel of RLVECN is constituted of two (2) distinct layers of FL, viz: Knowledge-Graph Embeddings and ConvNet [13].
(2) The high-quality data preprocessing techniques employed herein with respect to the benchmark data sets. We ensured that all constituent actors of a given social graph were transcoded to their respective discrete data representations, without any loss in semantics, and normalized prior to training and/or testing.

In conclusion, we intend to expand our experimentation scope to include more social network data sets and benchmark models. This research was made possible by International Business Machines (IBM), SHARCNET and Compute Canada (www.computecanada.ca).

References

1. Aggarwal, C.C. (ed.): Social Network Data Analytics. Springer, Boston (2011). https://doi.org/10.1007/978-1-4419-8462-3

2. Batagelj, V., Doreian, P., Ferligoj, A., Kejzar, N. (eds.): Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching. Visualization and Network Evolution. Wiley, Hoboken (2014)
3. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**, 1–113 (2009)
4. Deng, L.M., Yu, D.H.: Deep Learning: Methods and Applications. Foundations and trends in signal processing. Now Publishers (2014)
5. Goodfellow, I.G., Bengio, Y., Courville, A.C.: Deep learning. Nature **521**, 436–444 (2015)
6. Goodfellow, I.G., Bengio, Y., Courville, A.C. (eds.): Deep Learning. MIT Press, Cambridge (2017)
7. Gron, A. (ed.): Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media Inc., Newton (2017)
8. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016, pp. 855–864 (2016)
9. Hinton, G.E.: Learning multiple layers of representation. TRENDS Cognit. Sci. **11**(10), 428–433 (2007)
10. Hinton, G.E., et al.: Deep neural networks for acoustic modeling in speech recognition. IEEE Sig. Process. Mag. **29**, 82–97 (2012)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR), abs/1609.02907 (2017)
12. Krebs, V.E.: Organizational adaptability quotient. In: IBM Global Services (2008)
13. Molokwu, B.C.: Event prediction in complex social graphs using one-dimensional convolutional neural network. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI (2019)
14. Molokwu, B.C.: Event prediction in social graphs using 1-dimensional convolutional neural network. In: Canadian Conference on AI (2019)
15. Molokwu, B.C., Kobti, Z.: Event prediction in complex social graphs via feature learning of vertex embeddings. In: Gedeon, T., Wong, K.W., Lee, M. (eds.) Neural Information Processing, pp. 573–580. Springer, Cham (2019). https://doi.org/10.1007/978-1-4419-8462-3
16. Molokwu, B.C., Kobti, Z.: Spatial event prediction via multivariate time series analysis of neighboring social units using deep neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2019)
17. Namata, G., London, B., Getoor, L., Huang, B.: Query-driven active surveying for collective classification. In: Proceedings of the Workshop on Mining and Learning with Graphs, MLG-2012 (2012)
18. Patterson, J., Gibson, A. (eds.): Deep Learning: A Practitioner's Approach. O'Reilly Media Inc., Newton (2017)
19. Perozzi, B., Al-Rfou', R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, abs/1403.6652 (2014)
20. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015). http://networkrepository.com
21. Rozemberczki, B., Allen, C., Sarkar, R.: Multi-scale attributed node embedding (2019). arxiv: abs/1909.13021

22. Scott, J. (ed.): Social Network Analysis. SAGE Publications Ltd., Newbury Park (2017)
23. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Mag. **29**, 93–106 (2008)
24. Tabacof, P., Costabello, L.: Probability calibration for knowledge graph embedding models. In: International Conference on Learning Representations (ICLR), abs/1912.10000 (2020)
25. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web (2015)
26. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
27. Yang, S., Tian, J., Zhang, H., Yan, J., He, H., Jin, Y.: Transms: knowledge graph embedding for complex relations by multidirectional semantics. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI (2019)
28. Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., Qu, Y.: Multi-view knowledge graph embedding for entity alignment. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI. abs/1906.02390 (2019)
29. Zhao, B., Sen, P., Getoor, L.: Entity and relationship labeling in affiliation networks. In: Proceedings of the 23rd International Conference on Machine Learning (2006)