The Seybold Report

ISSN 1533-9211

DOI 10.17605/OSF.IO/C95XB

UNIFIED UNSUPERVISED DEEP LEARNING MODEL FOR PREDICTION OF TIME AND TOPIC-SPECIFIC INFLUENCERS FROM SOCIAL MEDIA

JOTHI P[1]* and PADMAPRIYA R[2]

[1, 2] School of Computer Studies, Rathnavel Subaramaniam College of Arts and Science, Coimbatore, Tamilnadu, India. *Corresponding Author Email: jothip_scsug@rvsgroup.com

Abstract

In Online Social Networks (OSNs), an efficient Influential User Prediction (IUP) is essential for different applications like sentiment analysis, online recommendation, etc. Among several prediction models, a Grey Wolf optimization with Graph Convolutional Neural Network (GW-GCNN) model can predict influencers by learning the latent vector representation of each netizen, including different centrality measures. But it models the netizen influence based on a fixed-size sub-network from the netizen's social action log and topic distributions, as well as was highly reliant on the different topics. It cannot learn an entire huge corpus since merely a limited part of the information was annotated by Ground Truth (GT). Hence, this article proposes a unified unsupervised GW-GCNN with the Long Short-Term Memory (LSTM) model without GT supervision. It aims to design the netizen's influence dynamics and discover the Influence Propagation (IPN) on multiple topics. The major contributions of this model are (i) measuring the time-aware and topic-related influences, (ii) modeling the IPN related to interval and topics using the Influence Attention-GCNN (IA-GCNN) that learns the netizen's latent vector representation under multiple topics and (iii) extracting temporal influence and learning the Influence Scores (ISs) by using a matrix-adaptive LSTM that considers the unsupervised objective. Moreover, the learned ISs of every topic are summed and max-pooled over a period to get every netizen's IS for predicting influencers. At last, the extensive experiments reveal that the GW-GCNN-LSTM achieves 93.9%, 92.5% and 92.4% accuracy for Facebook, Weibo and Twitter datasets, respectively during training, whereas it attains 94.1%, 93.5% and 94% accuracy for Facebook, Weibo and Twitter datasets, respectively during testing compared to the KSGC, Multi-view Influence (MvInf), InfACom-GCN and GW-GCNN algorithms.

Keywords: Online social networking, User influence, GW-GCNN, Temporal influence, LSTM, unsupervised learning

1. INTRODUCTION

OSNs such as Facebook, Twitter, etc., have seen a rise in popularity due to their real-time, transparency and quick data exchange qualities [1]. Academics are investigating features such as data exchange and burst topic recognition, as well as user influence analysis. Influence regions of members are not limited to the neighborhood and individuals can manipulate others through the data exchange mechanism [2]. Netizen engagement on an OSN is shown in Fig. 1. People from various locations can connect as part of the retweet networks, and the influence of a netizen on nearby retweeters is visible during the retweet operation, but less visible for distant non-neighboring retweeters [3]. Influencers are netizens who have the power to capture the interest of others. It is essential to recognize and forecast influencers in OSNs, as it offers a diverse set of potential. Influencers must be able to draw in more followers and broadcast their knowledge, and prediction techniques should be able to accurately depict user input [4].

ISSN 1533-9211

DOI 10.17605/OSF.IO/C95XB

Many studies have focused on calculating the effect of Twitter customers. Estimating influence enables the identification of netizens behavior that are critical for developing the system and offering remedies to relevant real-world tasks [5]. Academic evidence has metrics to calculate netizens influence, but in real-time OSNs it is difficult or expensive to get a full view of the information.

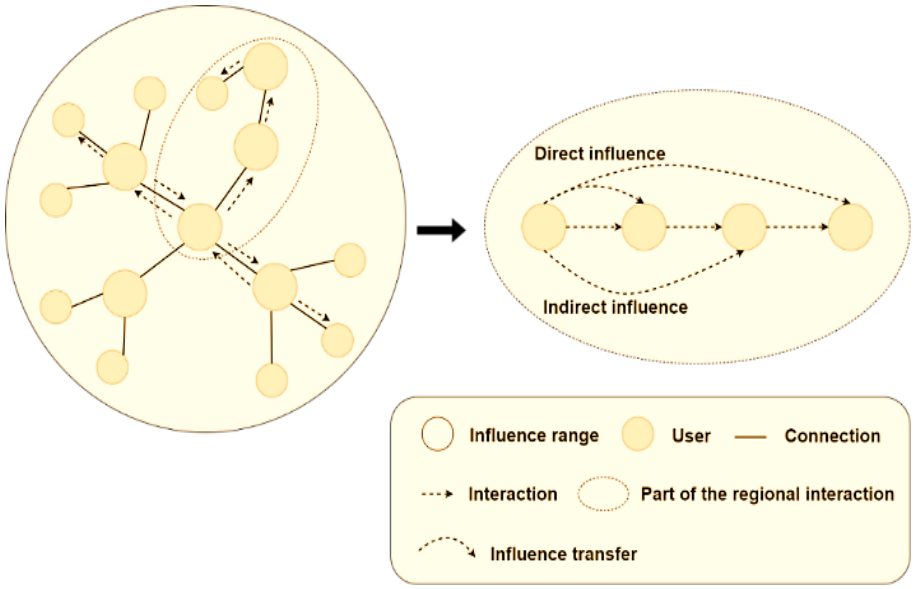

Figure 1: User collaboration in OSNs and virtual influence design

To combat these challenges, De Salve et al. [6] modeled the influencer prediction issue from the perspective of communities generated in OSNs and presented a technique that integrates centrality measures, data analytics and prediction schemes for IUP. Conversely, advanced feature selection schemes were needed with additional centrality measures for IUP. So, novel centrality measures were integrated with standard measures to fuse both node perspectives, while the most centrality measures were redefined for context and temporal dimensions. For this reason, Grey Wolf Optimization (GWO) [7-8] has been used to select measures from the temporal dimension of multiple periods, which were learned by a Convolutional Neural Network (CNN). However, only a limited group of IUs in OSNs were generated, which may affect the class distribution and accuracy of IUP. This issue was solved by the GCNN model, which learns the latent vector representation along with the different centrality measures and network topology for effective IUP [9]. However, the influence was only modeled based on fixed OSN topologies and topic allocations. Additionally, the nodes and edges in OSNs were numerous and changed rapidly, leading to supervised models that cannot learn complete huge corpora due to a limited part of the information being annotated with the GT.

Therefore, a unified unsupervised model is proposed in this manuscript to solve the above-mentioned issues in the supervised models by combining GCNN and LSTM networks. The main aim is to design the netizens influence dynamics and obtain the IPN on multiple topics because a netizens influence greatly depends on the topics. Initially, the time-aware and topic-

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

specific influences are measured. Then, the IA-GCNN is used to model the IPN process related to intervals and topics by learning multiple latent vector representations of the netizens. Moreover, a matrix-adaptive LSTM is adopted through optimizing the unsupervised fitness factor to capture the temporal influence and learn the ISs. Those learned ISs of each topic are summed and max-pooled over a period to get all netizens ISs for the final prediction. The prediction output can be the top-k netizens with the maximum ISs. Thus, the unsupervised objective function can train the GW-GCNN-LSTM model without control from the GT.

The residual sections are prepared as follows: Section 2 investigates related studies. Section 3 explains the GW-GCNN-LSTM for IUP and Section 4 displays its effectiveness. Section 5 concludes the study and recommends upcoming enrichments.

2. LITERATURE SURVEY

A new scheme called the Integrated Value of Influence (IVI) has been developed [10], which integrates the most significant topological features of the network to recognize its influential uses within it. But, its accuracy was not effective since it was not suitable for IUP in OSNs. A coreness-based VoteRank technique, namely NCVoteRank was developed [11] to discover spreaders (influential nodes) by considering the coreness value adjacent to the voting scheme. But, it needs additional metrics related to the network structures, tweets, etc., to increase the identification accuracy.

The detection of opinion leaders in data dissemination for the social network [12] was presented depending on the different centrality metrics of netizens like in-degree, out-degree and Betweenness. But, it considers only the netizens centrality values, whereas additional metrics were needed to increase the accuracy of detecting influencers. A community detection scheme depending on deep learning [13] was developed. The feature matrix was obtained by the deep sparse autoencoder and the low-dimensional feature matrix was clustered by the K-means algorithm to get the community structure. But, its accuracy was less since the low-dimensional feature matrix cannot define the topological information of the network when using a fewer number of compressed layers.

A new gravity model with effective distance [14] was presented to detect influential nodes according to the data merging and multi-level analysis. But its accuracy was less for large-scale networks. An improved gravity centrality measure [15] was developed based on the k-shell scheme called KSGC for detecting influential nodes in sophisticated OSNs. But its precision was less since it needs a weighted network with less time complexity. An influential nodes analysis scheme called LENC [16] was designed depending on the entropy and weight allocation of the edges linking it to determine the variance of edge weights and their impact on adjacent nodes. Conversely, the accuracy was not high for node influence sorting since it considers merely the impact of 1st and 2nd-order edges.

A deep learning model called Multi-view Influence (MvInf) prediction network [17] was developed by combining multi-view learning and graph attention neural network to predict netizen behavior. But it lacks accuracy in predicting the influence between heterogeneous

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

nodes since it was based on assumptions that netizens were only impacted by other netizens to define the association between netizens (homogeneous network).

A new technique called the Structure-based Identification Method (SIM) [18] was designed to detect the influential nodes depending on the system topology. However, several redundant elements were detected, which degrades the accuracy of creating a list of elements and their adjacent. Also, the splitting of influential nodes was complex while increasing the network dimension. An efficient algorithm called InfACom-GCN [19] was presented to identify the top-r k-influential groups in a large official network. But the mean accuracy was not effective and needs attention neural network to lessen the difficulty of the detection process.

In contrast with these previous researches, the proposed GW-GCNN-LSTM model can increase the accuracy of identifying influential netizens by learning various centrality metrics along with the time-aware topic-specific IS of each netizen in OSNs.

3. PROPOSED METHODOLOGY

In this section, the GW-GCNN-LSTM model is described in detail. Netizens data from OSNs is collected as a dataset, modeled and various centrality metrics extracted [7]. The GWO algorithm is applied to pick highly appropriate centrality metrics, which are provided to the unsupervised GCNN-LSTM for IUP. The notations utilized in this article are abridged in Table 1.

Table 1: Lists of notations

Notations	Explanation
N	Number of netizens
T	Total number of intervals
L	Number of interactions
G_t	Sequence of temporal attributed graph
V	Set of netizen nodes
A_t	Adjacency tensor
X_t	Netizen-topic affinity tensor
M	No. of topics in whole OSN
D	Topic embedding size
$X_{t(ij)}$	Term frequency of top D words in topic j
B	Time-aware and topic-related influence tensor
$\mathcal{N}_{i,t}$	Group of one-hop adjacent of node i at interval t
$e_{i,j}$	Attention coefficient
$GCNN_{\varphi}$	GCNN with parameters φ
$a_{i,j}$	Normalized attention coefficient
$F_t^{(0)}$	Initial layer features
$F_{t(i)}^{(p)}$	Output node representation
$\sigma(\cdot)$	ReLU or sigmoid activation function
$W^{(p)}$	Parameter matrix
F_t	Combined feature tensor

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

C_t	Centrality measure
I_t	Input gate of LSTM
G_t	Output gate of LSTM
P	Size of the LSTM's hidden states
$\mathcal{C}_t$	Cell state
O_t	Output gate of LSTM
H_t	Output state
$W_{x\cdot}, W_{h\cdot}, W_c$	Weights of LSTM layers
b_i, b_f, b_c, b_o	Biases of LSTM layers
T_W	Time window
$L(W, \lambda_l)$	Objective function
ζ_i	Tradeoff factor
$d^{T'}$	Documents
n_e	Learning epoch

3.1 Problem formation

Consider N number of netizens and all netizens have both text and interaction data. For instance, on Twitter, the text data comprises the set of tweets that are used to define netizens' affinity to specific topics. The interaction data is mined from the social actions between netizens like comments and retweets. If there are T periods and L kinds of interactions, then the social information is modeled as a series of time-based graphs, which are represented by$G_t = (V, A_t, X_t), t = 1, \dots, T$, where V denotes the group of netizens and$|V| = N$. The interaction data is modeled as the adjacency tensor $A_t = \mathbb{R}^{N \times N \times L}$ for L kinds of relations in t^{th} period. The netizen-topic affinity tensor $X_t = \mathbb{R}^{N \times M \times D}$ is the text data of N netizens in t^{th} period, M denotes the total topics in the whole OSN and D defines the topic embedding size.

According to this concept, the problem is formatted as follows: for the temporal attributed graphs $G_t = (V, A_t, X_t), t = 1, \dots, T$ that define the text and interaction data in OSNs, the aim is to obtain the time-aware and topic-related influence tensor $B \in \mathbb{R}^{N \times T \times M}$ for$V$.

3.2 Unified unsupervised GW-GCNN-LSTM model

Fig. 2 illustrates the unified unsupervised GW-GCNN-LSTM model for IUP. First, the time-based centrality measures are determined and the most relevant measures are chosen by the GWO. Textual and interactional data at each interval is also considered as input to the GCNN-LSTM model. For textual data, a Core-Concept Seeded Latent Dirichlet Allocation (CC-SeededLDA) [20] is used to implement topic extraction and find B in all intervals. For the time-based graphs of L kinds of relations, the IA-GCNN is applied to model the IPN and acquire multiple relations. Then, the time-based influence is obtained via updating the unsupervised fitness factor in the matrix-adaptive LSTM network

ISSN 1533-9211

DOI 10.17605/OSF.IO/C95XB

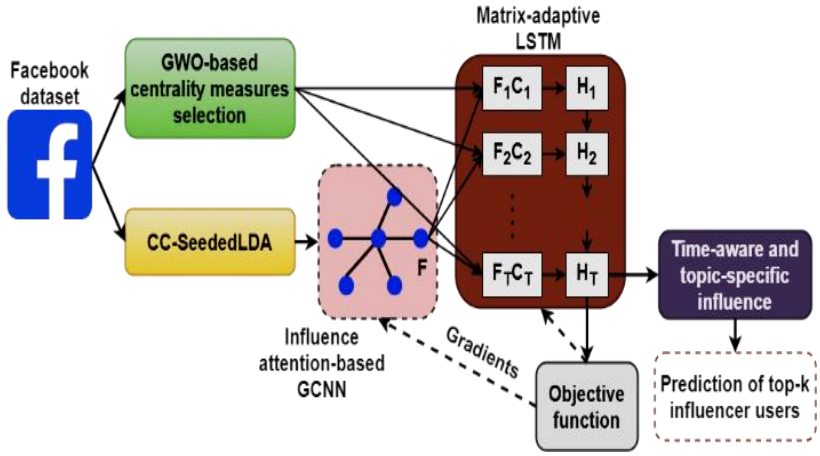

Figure 2: Workflow of unified unsupervised GW-GCNN-LSTM model for IUP

3.2.1. Topic extraction

In OSNs, a netizen normally has preferences in several topics. The topic extraction intends to get D-dimensional vector $X_{t(ij)}$ that defines the embedding of netizen i on topic j at inter val t. Thus, the data shared by a similar netizen in specific t is concatenated as unified text, providing $N \times T$ text. To find the topic interest of netizens, the CC-SeededLDA technique is used that can recognize latent topics in the following way:

1) **Unsupervised:** The text-topic allocation is achieved by the probability distribution with the Dirichlet prior.
2) **Supervised:** The CC-SeededLDA takes group of seed words as a descriptive of the original topics. In this manner, the text-topic allocation is obtained in certain fields.
3) **Online:** The model is progressively updated by earlier topic-word allocation as seed words to provide to CC-SeededLDA.

In all intervals, M topics are extracted and $X_t = \mathbb{R}^{N \times M \times D}, t = 1, \dots, T$ are obtained. For netizen i, $X_{t(ij)}$ indicates the occurrences of top-D words in j. A maximum component in X_t represents high interest that V on the resultant topic. The unsupervised CC-SeededLDA is appropriate for learning GW-GCNN-LSTM from scratch, where it recognizes the topics in posts. The supervised and online ways are suitable for online learning of the GW-GCNN-LSTM model.

3.2.2. Influence attention GCNN model

The IA net is constructed to formulate the IPN and get multiple relations. $A_t, t = 1, \dots, T$, related to L kinds of relations is obtained. Multiple kinds of relations have multiple roles to IPN. The GW-GCNN model considered a single kind of interaction (i.e., either comments or retweets of a netizen) or allocated a weight to relations based on field awareness. To solve this issue, this study proposes the IA-GCNN that fuses the node topic allocation with attention on the node's local neighborhood attributes and edges in multiple OSNs. The IA method concentrating on a certain i in G_t is defined without generalization loss. Consider $\mathcal{N}_{i,t}$ is the

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

group of single-hop adjacent of i at t. In this study, the attention factors are adopted for netizen-topics affinities and netizen-netizen relations as follows:

$$e_{i,j} = GCNN_{\varphi}\left(X_{t(i)}, X_{t(j)}, A_{t(ij:)}\right) \quad (1)$$

In Eq. (1), $j \in \mathcal{N}_{i,t}$, the attention coefficient $e_{i,j}$ determines the associative impact that netizen i has on netizen j and $GCNN_{\varphi}$ is the GCNN with variables φ. To allow netizens with various neighborhood dimensions, the coefficients are normalized with softmax as:

$$a_{i,j} = e^{(e_{i,j})} / \sum_{k \in \mathcal{N}_{i,t}} e^{(e_{i,k})} \quad (2)$$

In the IPN method, the OSN group broadcasts posts with several propagation cycles. So, the phenomena with several IA layers are modeled by combining the node's topic allocation vectors in their neighborhood. X_t is used as an entry node traits to the initial layer $\left(F_t^{(0)} = X_t\right)$. The p^{th} IA layer is implemented by

$$F_{t(i)}^{(p)} = \sigma\left(\sum_{j \in \mathcal{N}_{i,t}} a_{i,j} F_{t(j)}^{(p-1)} W^{(p)}\right) \quad (3)$$

In Eq. (3), $\sigma(\cdot)$ indicates the Rectified Linear Unit (ReLU) activation function, $F_{t(i)}^{(p)} \in \mathbb{R}^{N \times M \times d_n^{(p)}}$ defines the result node interpretations and $W^{(p)} \in \mathbb{R}^{d_n^{(p-1)} \times d_n^{(p)}}$ denotes the variable matrix. The combined feature tensor F_t from the result of the last IA layer defines the netizen-topic allocation after IPN.

3.2.3. Matrix-adaptive LSTM

Using $F_t, t = 1, \ldots, T$ and centrality measures $C_t, t = 1, \ldots, T$, a matrix-adaptive LSTM network is developed to get the time-aware and topic-related ISs for netizens. This LSTM is introduced to capture long-term dependencies, which certainly occur in temporal OSN information. The matrix-adaptive LSTM takes a series of matrices and provides the state matrices of each interval, operating as a many-to-many recurrent model.

The functions in a matrix-adaptive LSTM cell are defined by neglecting the size N as follows:

$$I_t = \sigma(C_t F_t W_{xi} + H_{t-1} W_{hi} + \mathcal{C}_{t-1} W_{ci} + b_i) \quad (4)$$

$$G_t = \sigma\left(C_t F_t W_{xf} + H_{t-1} W_{hf} + \mathcal{C}_{t-1} W_{cf} + b_f\right) \quad (5)$$

$$\mathcal{C}_t = G_t \odot \mathcal{C}_{t-1} + I_t \odot \tanh(C_t F_t W_{xc} + H_{t-1} W_{hc} + b_c) \quad (6)$$

$$O_t = \sigma(C_t F_t W_{xo} + H_{t-1} W_{ho} + \mathcal{C}_t W_{co} + b_o) \quad (7)$$

$$H_t = O_t \odot \tanh(\mathcal{C}_t) \quad (8)$$

In Eqns. (4) – (8), $\sigma(\cdot)$ is the sigmoid function, i.e., $\sigma(x) = {}^{1}/_{(1 + e^{-x})}$ and $I_t, G_t \in [0,1]^{M \times P}$ denote the input and output gates, where P indicates the dimension of the LSTM's hidden states and $\mathcal{C}_t \in \mathbb{R}^{M \times P}$ defines the cell state. The cell state acts as the data link between period $t-1$ and t. The input and forget gates with ranges regularized to $[0,1]$ assist the cell state regulator

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

how much data it must accept from the input (2nd term in Eq. (6)) and how much is inherited from the past period (initial term in Eq. (6)). $O_t \in [0,1]^{M\times P}$ denotes the output gate and $H_t \in \mathbb{R}^{M\times P}$ denotes the output state. The output gate filters data from C_t and forwards it to the output state that acts as LSTM's result.

Typically, the LSTM performs sequentially with $C_t F_t$ as the first input. C_t at t and H_t can be recurrently fed into the LSTM cell at $t+1$. The weights $W_{x\cdot} \in \mathbb{R}^{d_n^{(p)}\times P}, W_{h\cdot} \in \mathbb{R}^{P\times P}, W_{c\cdot} \in \mathbb{R}^{P\times P}$ and biases $b_i, b_f, b_c, b_o \in \mathbb{R}^P$ denote the network variables that are adjusted by backpropagation with the fitness factor. The influence tensor B is acquired from the aggregation of H_t after a max-pooling layer.

Algorithm 1 GW-GCNN-LSTM model training

Input: Facebook, Weibo and Twitter datasets

Result: Top-k influencers

1. **Begin**
2. Get the dataset having retweets, comments, likes and netizen details;
3. Apply data conversion on the training set to obtain multiple centrality measures of each netizen in different groups at different periods;
4. Pick the relevant centrality measures using the GWO algorithm;
5. Extract multiple topics using CC-SeededLDA and determine netizen-topic affinity tensor;
6. Create the IA-GCNN to model the IPN over periods and topics;
7. Obtain the latent vector representation of each netizen under different topics;
8. Define unsupervised fitness factor and construct a matrix-adaptive LSTM;
9. Learn temporal dependency, i.e., time-aware and topic-related ISs of netizens over different periods;
10. Sum and max-pool the learned IS to get all netizens ISs;
11. Predict the top-k influencers in a group;
12. Determine $L(W, \lambda_l)$ using Eq. (9) and update the weight matrices of the model;
13. **End**

This network can be adapted for online learning. At period T', this network trained at $T'-1$ is leveraged to determine the comprehensive $X_{T'}$ and the combined $F_{T'}$. The LSTM model is trained by initiating the variables (W) from earlier LSTM at $T'-1$. To extract the time-based correlation, a time window T_W is set as a hyperparameter: solely information attained at $[T'-T_W, T']$ is utilized to re-learn the LSTM, which enables the network to converge quicker than relearning from scratch.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

3.2.4. Fitness Factor

To determine the time-aware and topic-related influence, 3 hypotheses are considered while the unsupervised fitness factor is defined: (i) the netizens with a greater neighborhood and affinity must have a greater IS; (ii) active netizens are highly possible to contain a maximum IS compared to the idle netizens; (iii) the modification in the influence matrix must be smooth. According to these criteria, the absolute optimization dilemma is defined in Eq. (9), to get $B \in \mathbb{R}^{N\times T\times M}$,

$$\max L(W,\lambda_l) = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{t(ij:)}\left(1+\sum_{k=1}^{N} A_{t(jk:)}\right)\cdot \left\|B_{(it:)}\right\|^2 + \zeta_1 \sum_{t=1}^{T}\sum_{i=1}^{N}\left\|F_{t(i)}\right\|^2 \cdot \left\|B_{(it:)}\right\|^2 - \zeta_2 \sum_{t=2}^{T}\left\|B_{(:t:)} - B_{(:t-1:)}\right\|_F^2 \quad (9)$$

In Eq. (9), F_t denotes the combined netizen-topic affinity tensor in period t, W has the weight matrices in the LSTM and IA-GCNN and $\zeta_i > 0, i = 1,2$ are the tradeoff factors to equilibrium 3 elements. The higher value of $B_{(itm)}$ defines that i has a greater influence on topic m at t. A restraint is included to regularize netizen ISs on a topic for all periods. The Back-Propagation through Time (BPTT) scheme is used for model training and learning the netizen ISs.

So, the GW-GCNN-LSTM model is extended to an online manner using the IA-GCNN and matrix-adaptive LSTM networks. The pseudocode for the online learning of the GW-GCNN-LSTM model is given in Algorithm 2.

The revised objective function is defined using T' information received,

$$\max L(W,\lambda_l) = \sum_{t=T'-T_W}^{T'}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{t(ij)}\left(1+\sum_{k=1}^{N} A_{t(jk)}\right)\cdot \left\|B_{(it:)}\right\|^2 + \zeta_1 \sum_{t=T'-T_W}^{T'}\sum_{i=1}^{N}\left\|F_{t(i)}\right\|^2 \cdot \left\|B_{(it:)}\right\|^2 - \zeta_2 \sum_{t=T'-T_W+1}^{T'}\left\|B_{(:t:)} - B_{(:t-1:)}\right\|_F^2 \quad (10)$$

Moreover, the learned IS of each topic is summed and max-pooled over a period to get each netizen's IS, which is used to recognize top-k influencers in the labeled datasets.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

Algorithm 2 Online learning of GW-GCNN-LSTM model

Input: $A_t^{(l)}, F_t\ (t = T' - T_W, \dots, T')$, texts $d^{T'}$, earlier CC-SeededLDA$(T'-1)$, earlier LSTM$(T'-1)$, learning epoch $n_e$and hyperparameters a, ζ_1, ζ_2, T_W

1. **Begin**
2. Get the topic-word allocation from CC-SeededLDA$(T'-1)$ as seed dissemination for CC-SeededLDA(T');
3. Train CC-SeededLDA(T') using $d^{T'}$;
4. Determine $X_{T'(ij)}$;
5. Determine $F_{T'}$ using Eqns. (1) – (3);
6. Obtain W from LSTM$(T'-1)$ to LSTM(T');
7. $\boldsymbol{for}(epoch = 1; epoch \leq n_e)$
8. $\boldsymbol{for}(t = T' - T_W; t \leq T')$
9. Determine I_t, G_t, C_t, O_t, H_t using Eqns. (4) – (8);
10. ***end for***
11. Calculate $L(W, \lambda_l)$ using Eq. (10);
12. Back-propagate and modify W;
13. ***end for***
14. **End**

4. RESULTS AND DISCUSSIONS

The effectiveness of the GW-GCNN-LSTM model is assessed by implementing it using Python code. Every experiment is executed on a machine with a quad-core Intel i5 2.20 GHz processor and 64 GB memory.

4.1 Dataset description

1. Facebook data from the Kaggle website is obtained, which comprises 4 .csv documents such as post.csv, comment.csv, like.csv and member.csv from the different open Facebook communities [21].
2. Weibo dataset: It is the most well-known OSN. The Weibo dataset has 1776950 netizens [22]. This study randomly selects 50000 examples to predict influential netizens.
3. Twitter: It is a well-known OSN. This dataset includes 456626 netizens [23]. In this study, 50000 examples are randomly chosen for IUP.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

4.2 Parameter Settings

The proposed GW-GCNN-LSTM samples the sub-net with a random walk with an activation probability of 0.75 and a predetermined dimension of 60. The vector size is 64 and a 64-dimensional net embedding vector is pre-learned by auto-encoder schemes. The 3-layer GCNN architecture is used for training, with 64 hidden neurons in the initial and second levels, and 2 hidden neurons in the output level. 60% and 40% examples of sampling data are chosen for learning and testing, and the mini-batch dimension is assigned to 512. In the influence dissemination model, 5000 simulations are considered after choosing a new netizen and including the seed set, and the observation window is set from 1-10-2022 to 28-2-2023.

The matrix-adaptive LSTM is trained by the Adam optimizer with a training rate of 0.0001, β_1 is 0.9 and β_2 is 0.999. The tradeoff factors in Eq. (9) (ζ_1, ζ_2) are examined from 10^{-4} to 10^{4} with a step of 10^{1}. The weight matrices in the GCNN-LSTM model are initialized by Xavier initialization [24]. Table 2 lists the parameters set for existing models: GW-GCNN [9], KSGC [15], MvInf [17] and InfACom-GCN [19] to compare the IUP efficacy.

Table 2: Parameter settings for existing models

Algorithms	Parameters	Range
KSGC [15]	No. of nodes	1000
	Mean degree	50
	Maximum degree	550
	Mixing variable of the community structure	0.7
MvInf [17]	Learning rate	0.01
	Weight decay	$5e^{-5}$
	Dropout rate	0.5
InfACom-GCN [19]	Training rate	0.1
	Dropout rate	0.2
	No. of epoch	100
GW-GCNN [9]	Population size of grey wolf	100
	Maximum iteration	250
	GCN hidden neurons	64
	GCN dropout rate	0.5
	No. of GCN layers	3
	GCN activation function	ReLU
	Loss function	Cross-entropy
	Batch size	512
	No. of epoch	5000
	Initial learning rate	0.5
	Learning rate decay	0.95

The existing models are also tested using the given Facebook, Weibo and Twitter databases and evaluated to realize the efficiency of the proposed GE-GCNN-LSTM model.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

4.3 Evaluation metrics

The IUP performance is evaluated by different metrics given below.

- **Precision:** It determines the number of proper positive predictions achieved.

$$Precision = \frac{True\ Positive\ (TP)}{TP+False\ Positive\ (FP)} \quad (11)$$

- **Recall:** It determines the number of proper positive predictions achieved over each positive prediction that could have been achieved.

$$Recall = \frac{TP}{TP+False\ Negative\ (FN)} \quad (12)$$

- **F-measure:** It is defined by

$$F-measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (13)$$

- **Accuracy:** It is calculated according to the variance between the predicted and expected values.

$$Accuracy = \frac{TP+True\ Negative\ (TN)}{TP+TN+FP+FN} \quad (14)$$

4.4 Performance analysis for training phase

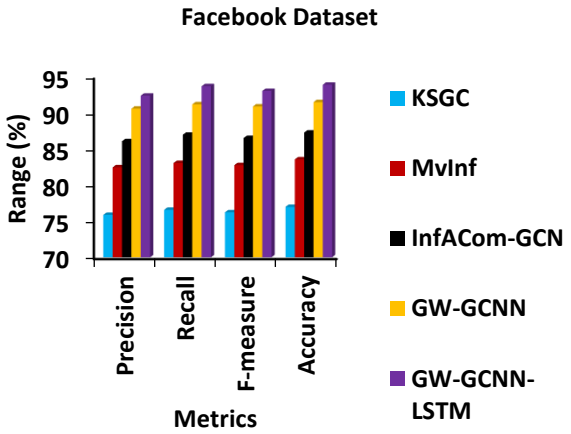

Figure 3: Analysis of various IUP models on Facebook dataset during training

In Fig. 3, comparison of proposed and existing IUP models on the Facebook dataset is validated in terms of different metrics. It is shown that the precision of GW-GCNN-LSTM model is increased by 21.74%, 12%, 7.32% and 1.99% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The recall of GW-GCNN-LSTM model is increased by 22.32%, 12.76%, 7.7% and 2.74% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The f-measure of GW-GCNN-LSTM model is increased by 22.03%, 12.38%, 7.51% and 2.37% compared to the KSGC, MvInf, InfACom-GCN and GW-

ISSN 1533-9211

DOI 10.17605/OSF.IO/C95XB

GCNN models, respectively. Also, the accuracy of GW-GCNN-LSTM model is increased by 21.95%, 12.32%, 7.56% and 2.62% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively.

Fig. 4 portrays a comparison of proposed and existing IUP models when training by the Weibo dataset in terms of different metrics. It is shown that the GW-GCNN-LSTM increases the precision by 16.05%, 11.64%, 7.18% and 2.47% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The GW-GCNN-LSTM model increases the recall by 16.1%, 12.1%, 7.4% and 3.2% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The GW-GCNN-LSTM increases the f-measure by 16.06%, 11.95%, 7.37% and 2.91% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. Also, the GW-GCNN-LSTM increases the accuracy by 15.63%, 11.85%, 7.43% and 2.78% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively.

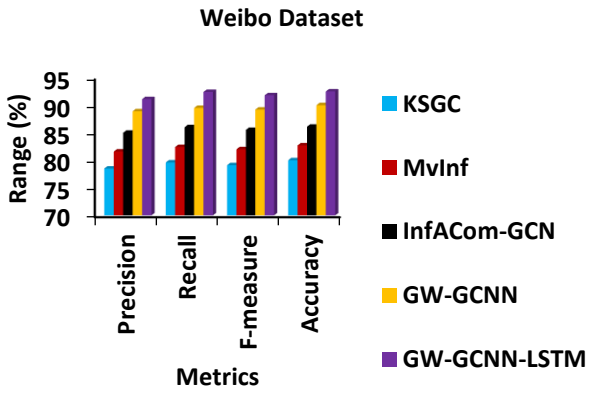

Figure 4: Analysis of various IUP models on OAG dataset during training

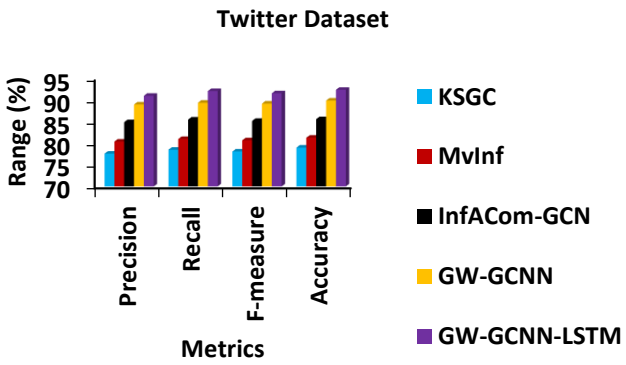

Figure 5: Analysis of various IUP models on Twitter dataset during training

ISSN 1533-9211

DOI 10.17605/OSF.IO/C95XB

In Fig. 5, comparison of proposed and existing IUP models when training by the Twitter dataset is shown in terms of different metrics. It is shown that the precision of GW-GCNN-LSTM model is increased by 17.27%, 13.18%, 7.18% and 2.25% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The recall of GW-GCNN-LSTM model is increased by 17.32%, 13.7%, 7.72% and 3.02% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The f-measure of GW-GCNN-LSTM model is increased by 17.29%, 13.51%, 7.51% and 2.69% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. Also, the accuracy of GW-GCNN-LSTM model is increased by 16.96%, 13.65%, 7.94% and 2.78% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively.

4.5 Performance analysis for testing phase

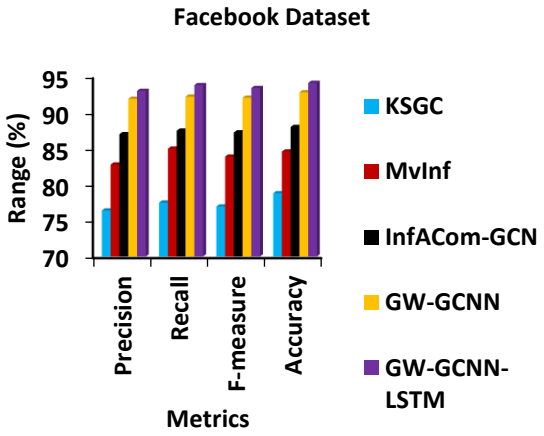

Figure 6: Analysis of various IUP models on Facebook dataset during testing

Fig. 6 portrays the comparison of proposed and existing IUP models when testing by the Facebook dataset. It is observed that the precision of GW-GCNN-LSTM model is improved by 21.7%, 12.3%, 6.9% and 1.2% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively. The recall of GW-GCNN-LSTM model is improved by 21.03%, 10.4%, 7.2% and 1.7% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The f-measure of GW-GCNN-LSTM model is enhanced by 21.4%, 11.3%, 7.05% and 1.47% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively. Also, the accuracy of GW-GCNN-LSTM model is improved by 19.42%, 11.23%, 6.93% and 1.4% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. Accordingly, the GW-GCNN-LSTM model outperformed other IUP models in both training and testing phases by learning time-aware and topic-specific influence of each netizen over a period without GT supervision.

In Fig. 7, a comparison of proposed and existing IUP models is illustrated when testing by the Weibo dataset. It is observed that the precision of GW-GCNN-LSTM model is 23.2%, 13.51%, 7.44% and 3.7% higher than the KSGC, MvInf, InfACom-GCN and GW-GCNN models,

ISSN 1533-9211

DOI 10.17605/OSF.IO/C95XB

respectively. The recall of GW-GCNN-LSTM model is 22.05%, 13.28%, 7.02% and 3.33% higher than the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively. The f-measure of GW-GCNN-LSTM model is 22.62%, 13.33%, 7.17% and 3.46% higher than the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. Also, the accuracy of GW-GCNN-LSTM model is 22.54%, 13.2%, 7.47% and 3.66% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. Thus, the GW-GCNN-LSTM can perform both training and testing phases efficiently compared to the existing IUP models by learning time-aware and topic-specific influence of each netizen.

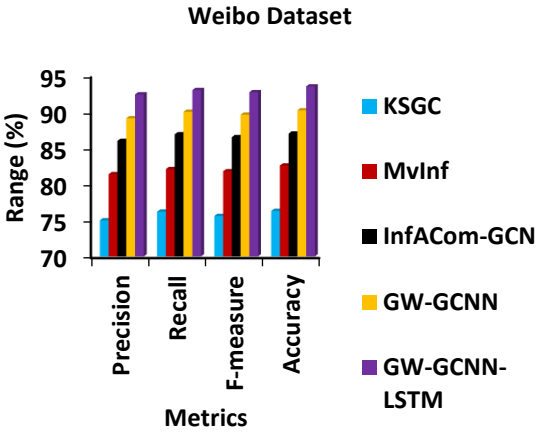

Figure 7: Analysis of various IUP models on OAG dataset during testing

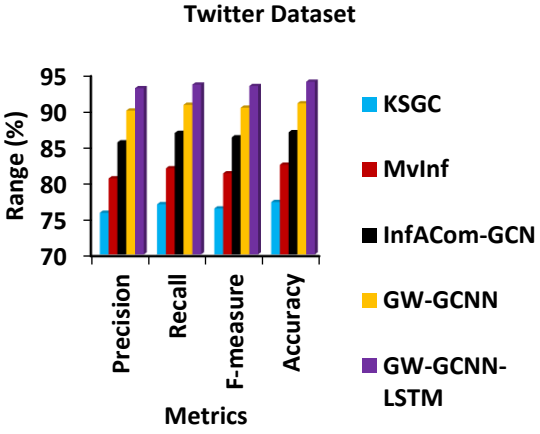

Figure 8: Analysis of various IUP models on Twitter dataset during testing

Fig. 8 portrays the comparison of proposed and existing IUP models when testing by the Twitter dataset. It is observed that the precision of GW-GCNN-LSTM is improved by 22.82%, 15.51%, 8.76% and 3.44% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively. The recall of GW-GCNN-LSTM is improved by 21.6%, 14.2%, 7.7% and 3.1%

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN models, respectively. The f-measure of GW-GCNN-LSTM is enhanced by 22.3%, 14.9%, 8.23% and 3.32% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively. Also, the accuracy of GW-GCNN-LSTM is improved by 21.6%, 13.9%, 8.1% and 3.3% compared to the KSGC, MvInf, InfACom-GCN and GW-GCNN, respectively. So, the GW-GCNN-LSTM model outperformed other IUP models in both training and testing phases by learning time-aware and topic-specific influence of each netizen over a period.

5. CONCLUSION

This study developed a unified unsupervised GW-GCNN-LSTM model for IUP. It used the IA-GCNN and matrix-adaptive LSTM to get the time-aware and topic-related ISs of each netizen at different periods. The learned ISs of every topic were summed and max-pooled over a period to obtain all netizen's ISs. Based on the final IS, the top-k influencers in the OSNs were predicted. Finally, the simulation results proved that the GW-GCNN-LSTM has a 94.1%, 93.5% and 94% accuracy for Facebook, Weibo and Twitter datasets, respectively compared to the other models for IUP in OSNs.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Methodology, Software, Validation, Jothi; Formal analysis, Investigation, Padmapriya; Resources, Data duration, Writing—Original draft preparation, Jothi; Reviewing and Editing, Jothi; Visualization, Supervision, Padmapriya.

References

1) N. A. Ghani, S. Hamid, I. A. T. Hashem, and E. Ahmed, "Social Media Big Data Analytics: A Survey", *Computers in Human Behavior*, Vol. 101, pp. 417-428, 2019.

2) A. Arora, S. Bansal, C. Kandpal, R. Aswani, and Y. Dwivedi, "Measuring Social Media Influencer Index-Insights from Facebook, Twitter and Instagram", *Journal of Retailing and Consumer Services*, Vol. 49, pp. 86-101, 2019.

3) Y. Yan, F. Toriumi, and T. Sugawara, "Understanding How Retweets Influence the Behaviors of Social Networking Service Users via Agent-Based Simulation", *Computational Social Networks*, Vol. 8, No. 1, pp. 1-21, 2021.

4) D. Weber, and F. Neumann, "Amplifying Influence through Coordinated Behaviour in Social Networks", *Social Network Analysis and Mining*, Vol. 11, No. 1, pp. 1-42, 2021.

5) K. Jamil, L. Dunnan, R. F. Gul, M. U. Shehzad, S. H. M. Gillani, and F. H. Awan, "Role of Social Media Marketing Activities in Influencing Customer Intentions: A Perspective of a New Emerging Era", *Frontiers in Psychology*, Vol. 12, pp. 1-12, 2022.

6) A. De Salve, P. Mori, B. Guidi, L. Ricci, and R. D. Pietro, "Predicting Influential Users in Online Social Network Groups", *ACM Transactions on Knowledge Discovery from Data*, Vol. 15, No. 3, pp. 1-50, 2021.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/C95XB

7) P. Jothi, and R. Padmapriya, "In Online Social Network using Grey Wolf and Deep Learning Technique for Influential User Prediction (IUP). *Journal of Pharmaceutical Negative Results*, Vol. 13, No. 10, pp. 586-599, 2022.

8) P. Jothi, and R. Padmapriya, "An Influential User Prediction in Social Network Using Centrality Measures and Deep Learning Method. In: *Proc. of Data Analytics and Management*, Springer Nature, Singapore, pp. 813-829, 2023.

9) P. Jothi, and R. Padmapriya, "Improvements in Deep Learning for Predicting Influential Users on Social Networks",

10) A. Salavaty, M. Ramialison, and P. D. Currie, "Integrated Value of Influence: An Integrative Method for the Identification of the Most Influential Nodes within Networks", *Patterns*, Vol. 1, No. 5, pp. 1-14, 2020.

11) S. Kumar, and B. S. Panda, "Identifying Influential Nodes in Social Networks: Neighborhood Coreness Based Voting Approach", *Physica A: Statistical Mechanics and its Applications*, Vol. 553, pp. 1-21, 2020.

12) A. U. Rehman, A. Jiang, A. Rehman, A. Paul, and M. T. Sadiq, "Identification and Role of Opinion Leaders in Information Diffusion for Online Discussion Network", *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-13, 2020.

13) S. Li, L. Jiang, X. Wu, W. Han, D. Zhao, and Z. Wang, "A Weighted Network Community Detection Algorithm Based on Deep Learning. *Applied Mathematics and Computation*, Vol. 401, pp. 1-9, 2021.

14) Q. Shang, Y. Deng, and K. H. Cheong, "Identifying Influential Nodes in Complex Networks: Effective Distance Gravity Model", *Information Sciences*, Vol. 577, pp. 162-179, 2021.

15) X. Yang, and F. Xiao, "An Improved Gravity Model to Identify Influential Nodes in Complex Networks Based on K-Shell Method", *Knowledge-Based Systems*, Vol. 227, pp. 1-11, 2021.

16) B. Wang, J. Zhang, J. Dai, and J. Sheng, "Influential Nodes Identification Using Network Local Structural Properties", *Scientific Reports*, Vol. 12, No. 1, pp. 1-13, 2022.

17) H. Xu, B. Jiang, and C. Ding, "MvInf: Social Influence Prediction with Multi-View Graph Attention Learning", *Cognitive Computation*, pp. 1-7, 2022.

18) T. Wang, P. Zeng, J. Zhao, X. Liu, and B. Zhang, "Identification of Influential Nodes in Industrial Networks Based on Structure Analysis", *Symmetry*, Vol. 14, No. 2, pp. 1-14, 2022.

19) N. A. Hussein, H. M. Mokhtar, and M. E. El-Sharkawi, "Influential Attributed Communities via Graph Convolutional Network (InfACom-GCN)", *Information*, Vol. 13, No. 10, pp. 1-17, 2022.

20) H. Huang, M. Harzallah, F. Guillet, and Z. Xu, "Core-Concept-Seeded LDA for Ontology Learning", *Procedia Computer Science*, Vol. 192, pp. 222-231, 2021.

21) https://www.kaggle.com/mchirico/cheltenham-s-facebook-group?select=database.sqlite.zip

22) J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing, "Who Influenced You? Predicting Retweet via Social Influence Locality", *ACM Transactions on Knowledge Discovery from Data*, Vol. 9, No. 3, pp. 1-26, 2015.

23) M. De Domenico, A. Lima, P. Mougel, and M. Musolesi, "The Anatomy of a Scientific Rumor", *Scientific Reports*, Vol. 3, No. 1, pp. 1-9, 2013.

24) X. Glorot, and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks", In: *Proc. of the Thirteenth International Conf. On Artificial Intelligence and Statistics*, pp. 249-256, 2010.