Influenza-like Symptom Recognition using Mobile Sensing and Graph Neural Networks

Guimin Dong, Lihua Cai, Debajyoti Datta, Shashwat Kumar, Laura E. Barnes, Mehdi Boukhechba
{gd5ss,lc3cp,dd3ar,sk9epp,lb3dp,mob3f}@virginia.edu
University of Virginia
Charlottesville, VA, USA

Abstract

Early detection of influenza-like symptoms can prevent widespread flu viruses and enable timely treatments, particularly in the post-pandemic era. Mobile sensing leverages an increasingly diverse set of embedded sensors to capture fine-grained information of human behaviors and ambient contexts, and can serve as a promising solution for influenza-like symptom recognition. Traditionally, handcrafted and high level features of mobile sensing data are extracted by manual feature engineering and convolutional/recurrent neural network respectively. In this work, we apply graph representation to encode the dynamics of state transitions and internal dependencies in human behaviors, leverage graph embeddings to automatically extract the topological and spatial features from graph inputs, and propose an end-to-end graph neural network (GNN) model with multi-channel mobile sensing input for influenza-like symptom recognition based on people's daily mobility, social interactions, and physical activities. Using data generated from 448 participants, we show that GNN with GraphSAGE convolutional layers significantly outperforms baseline models with handcrafted features. Furthermore, we use GNN interpretability method to generate insights (e.g., important nodes and graph structures) about the importance of mobile sensing for recognizing Influenza-like symptoms. To the best of our knowledge, this is the first work that applies graph representation and graph neural network on mobile sensing data for graph-based human behavior modeling and health symptoms prediction.

CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**; • **Computing methodologies → Neural networks**.

Keywords

Mobile Sensing, Graph Neural Network, Influenza-like Symptom Recognition, Behavior Modeling, Graph Representation Learning

ACM Reference Format:
Guimin Dong, Lihua Cai, Debajyoti Datta, Shashwat Kumar, Laura E. Barnes, Mehdi Boukhechba. 2021. Influenza-like Symptom Recognition using Mobile Sensing and Graph Neural Networks. In *ACM Conference on Health, Inference, and Learning (ACM CHIL '21), April 8–10, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3450439.3451880

ACM CHIL '21, April 8–10, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8359-2/21/04…$15.00
https://doi.org/10.1145/3450439.3451880

1 Introduction

From Spanish flu to SARS, to swine flu, global pandemics caused by influenza viruses have devastated human society, leading to significant loss in human lives and economies [57]. These pandemics have put the established public health and socioeconomic systems under the microscope, and resulted in many policy changes that were meant for combating future pandemics [47]. However, the ongoing COVID-19 pandemic reveals unseen vulnerabilities of the current systems in containing the virus due to its delayed appearance of symptoms and easy contagion, and poses much greater challenges to governments' public health response [5]. In the United States, the Center for Disease Control and Prevention (CDC) initialized the U.S. Influenza Surveillance System to collect and analyze influenza-related information and monitor influenza activities [1]. Aggregated outpatient data about "influenza-like illness" (ILI) are collected and reported hierarchically from local hospitals and public health systems to CDC [37]. Due to rapid expansion of influenza viruses, the effectiveness of these public health interventions can be delayed and diminished. Thus, it is critical to develop intelligent influenza surveillance systems to continuously monitor influenza activity, automatically detect early ILI among population, and accurately predict influenza outbreaks [26].

There are multiple established research focusing on influenza activity monitoring and influenza-like symptom recognition. Forsad et al. tested a contactless syndromic surveillance platform consisted of a microphone array and a thermal camera, installed in public waiting area of a university hospital, to continuously and passively monitor influenza infection scenes by characterizing the captured influenza bio-clinical signals [1]. The trends of Google search and Wikipedia pageviews on influenza related terms, and ILI linguistic signals extracted from social media platforms are used to build natural language processing models to predict influenza dynamics[49]. Multiple data sources, including air quality data and insurance data, have been leveraged to predict the probability of influenza outbreaks [58]. Even though the above methods can detect ILI and predict influenza trends in promising ways, their estimations have certain limitations. For example, Google Flu Trend, which predicts flu activity by analyzing search queries, has been discontinued due to its inaccuracies and lack of reproducibility [39]. Furthermore, installing edge devices in hospitals to capture influenza signal could lead to biased estimations that can not be generalized to the larger population [1].

[1]CDC-Flu Activity & Surveillance https://www.cdc.gov/flu/weekly/index.htm.

Mobile sensing provides a potential solution for ILI recognition and influenza surveillance through continuously and unobtrusively collecting both behavioral and physiological signals generated from human users [6]. Embedded sensors in mobile devices (e.g., personal smartphones and smartwatches), such as accelerometer, GPS sensor, and Bluetooth sensor, have been applied to monitor human behaviors and track daily activities. The resulted data from these embedded sensors can be used to infer human health status, monitor mental health states, and deliver medical interventions [7, 10, 12]. Traditionally, handcrafted features are extracted from mobile sensing data to analyze human behavior patterns. For example, motion features such as magnitude of acceleration, can be extracted from accelerometer to study their correlations with different user contexts (e.g., location, activity, social context) [8]. Instead of using handcrafted features, which are usually based on heuristics and domain knowledge, high level complex features can be automatically extracted from mobile sensing data, and leveraged to improve generalization of predictive modeling using deep learning algorithms. For example, Boukhechba et al. built convolutional neural network (CNN) models based on photoplethysmogram (PPG) data to infer ambulatory activities [9]. However, traditional handcrafted feature engineering and general deep learning methods (e.g., CNNs) are limited for multi-modal mobile sensing data. On one hand, low level features extracted by handcrafted feature engineering can miss information about the inter-dependencies in different sensing modalities. On the other hand, the complex interactions within data points in and among different sensing streams can not be naturally encoded in deep features by CNN, especially when the data have hidden structural patterns and are generated from non-Euclidean domains [54]. Given that some discrete sensor signals, such as GPS and Bluetooth streams, can be naturally represented as graphs, which encode the structural interdependence and information generated from non-euclidean spaces, we proposed an end-to-end graph neural network (GNN) framework to infer the existence of influenza-like symptoms based on people's multi-modal daily mobile sensing data.

Our contributions in this work are summarised below:

- We propose an end-to-end Graph neural network framework to model human behaviors by leveraging multi-modal mobile sensing data. The goal is to automatically extract high level features representing the dynamic interactions between human states to predict symptoms of ILI. To our best knowledge, this is **the first work that applies Graph neural networks to infer human health states using mobile sensing.** This end-to-end GNN framework can be easily generalized and applied in other mobile sensing tasks, such as monitoring mental health symptoms [2, 11, 13, 21].
- We demonstrate the performance of the proposed GNN framework by applying our framework on a large mobile sensing dataset collected **in the wild** to perform influenza-like symptom recognition. The results show that the GNN models significantly outperform baseline methods using handcrafted features.
- Instead of using handcrafted features to characterize the nodes in input graphs, we apply a **graph representation learning method, called node embedding, to automatically generate topological embeddings of nodes.** Our results imply that graph representation learning methods can be leveraged to generate embeddings to represent each individual human state.
- To enhance trust with the 'black box' GNNs, we apply interpretable GNN methods to increase **model interpretability**. We illustrate the transparency of the GNN models such that this framework can be better understood by policy decision makers and medical professionals.

The remainder of this work is presented as follows. In Section 2, we summarize recent works on influenza-like symptoms recognition and the benefit of applying GNN to mobile sensing data. In Section 3, we present how we model human behaviors by using mobile sensing data, and introduce graph-based behavior modeling by using graph representation learning and GNNs. In Section 4, we introduce our mobile sensing data and summarizes them using graph statistics. Section 5 presents the results from our case study. We end the current work with some conclusion marks and discussions of future works in Section 6.

2 Related Work

Several researchers have previously built machine learning models to predict influenza-like symptoms by analyzing human mobility behaviors captured by GPS sensors. Barlacchi et al. extracted fine-grained mobility behavior markers such as radius of gyration and number of unique visited places from GPS trajectories to predict future presence of influenza-like symptoms [4]. However, these handcrafted behavior markers require domain knowledge, and can only serve as low level representations of human mobility behaviors. The complex relationships between human mobility states cannot be fully captured by using these handcrafted features. For example, the relationship between place visits cannot be encoded by using these handcrafted features (e.g., transitioning from one place to another), nor can they capture the topological information of the mobility traces. Similarly, by analyzing human mobility behaviors, Ma et al. built an ensemble Support Vector Machine (SVM) to improve the generalization using handcrafted features [37]. Even though the ensemble SVM algorithm produces promising results in predicting influenza-like symptoms, there is still room for improvement in prediction performance by using more sophisticated feature engineering methods to capture complex human mobility behavior patterns. In our current work, we propose to leverage GNNs to automatically decode complex signals in mobile sensing data for influenza-like symptom recognition.

In addition to mobility, people with influenza-like symptoms may also display different physical and social behaviors [43]. Bluetooth sensors have been leveraged to approximate people's social interactions [51], while motion data from accelerometer can be leveraged to recognize physical activities [30]. We found no existing works, which applied Bluetooth encounter and physical activity data together with mobility data to infer influenza-like symptoms. Given that these multi-modal data can be represented as graphs based on their underlying temporal relationships, we propose to combine them using a multi-modal approach, aiming to improve prediction performance.

Graph representation learning has been applied to encode various data in machine learning tasks to solve several real world

problems such as social network recommendation [18], drug-to-drug interaction prediction [46], and knowledge reasoning [38]. Authors in these works have applied graph representation to capture complex interdependent relationships between individual entities in different problems. Graph neural networks can automatically extract graph structural information and capture composite interconnections from graph-structured data. For instance, in the graph representation of road networks, road segments represent their spatial information; and edges linking two adjacent segments represent the connectivity between them [50]. The authors applied GNNs to predict traffic flow by learning spatiotemporal patterns hidden in the complex road networks. Road network embeddings are learned by using GNNs to generate semantic representation of road segments such that the structural functions road networks can be leveraged for transportation optimization [53]. In social network, Fan et al. encoded the user-to-user and user-to-item relationships using graphs and applied GNNs to make social recommendations based on the social interactions between users [18]. In this work, we apply different GNN algorithms with multiple types of node embedding techniques to recognize influenza-like symptoms using multi-modal mobile sensing data.

3 Methodology

In this section, we present our methodology for comparing handcrafted feature engineering and graph representation methods to model human behaviors, and infer whether people suffer from at least one influenza-like symptoms or not given their daily mobile sensing observations. Influenza-like symptoms refer to fever, feeling feverish/chills, cough, sore throat, runny or stuffy nose, muscle or body aches, headaches, fatigue [2]. Existence of influenza-like symptoms could decrease people's daily mobility, social interactions, and physical activeness. For example, when people feel fatigue or muscle ache, they may subconsciously become more sedentary and avoid unnecessary traveling, even when they do not realize these early stage flu symptoms. Significant reductions of time spent in daily social interaction and average duration per contact were observed for participants who had influenza-like illness in [43]. We hypothesize that **people's mobility behaviors, social interactions, and physical activities are correlated with influenza-like symptoms**, and can be used to accurately recognize them.

We use GPS sensors to capture human mobility behaviors by tracking people's daily travel patterns; Bluetooth sensors, which record Bluetooth devices in proximity, to approximate social interactions; and activity data derived from accelerometer and gyroscope data to represent physical activities. For mobility data, we applied DBSCAN [17] to cluster GPS coordinates into visited places, and form sequences of daily place visits. Activities (i.e $in-vehicle, running, walking$) were recognized by using Google Activity Recognition API [22] for Andriod devices and CMMotion-Activity API for iOS devices [3]. We denoted daily place visit trace as $\mathbf{X_p}$, daily Bluetooth encounter traces as $\mathbf{X_b}$, and daily activity trace as $\mathbf{X_a}$. Our prediction task can be generalized as learning a function $\mathcal{F}(\bullet|\theta)$ given the inputs of $\mathbf{X_p}, \mathbf{X_b}, \mathbf{X_a}$ to predict existence of influenza-like symptoms denoted as $y \in \{0, 1\}$, such that $\mathcal{H}(y, \hat{y})$ can be minimized, where $\mathcal{H}(\bullet)$ is the cross entropy loss function.

[2]CDC-Flu symptoms https://www.cdc.gov/flu/symptoms/symptoms.html.

Behavior	Feature	Description
Mobility	radius_gyration	radius of gyration
	number_visits	total number of places visited
	max_distance	maximum distance of place visited from centroid
	travel_distance	total distance traveled
	max_distance_home	maximum distance of place visited from home
	travel_distance_std	standard deviation of distance of trips
	entropy_visits	randomness of places visited
	unique_visits	number of unique place
	multiple_place	number of placed visited at least twice
Social Interaction	num_encounter	number of Bluetooth devices detected
	entropy_encounter	randomness of Bluetooth devices
	unique_encounter	number of unique Bluetooth devices detected
Physical Activity	act_entropy	randomness of physical activity

Table 1: Description of Handcrafted Features

We propose to compare two feature engineering methods: 1) using handcrafted feature engineering as shown in section 3.1; and 2) using Graph representation to automatically extract high level features in section 3.2.

3.1 Handcrafted Feature Engineering

To measure human mobility and build a predictive model for influenza-like symptom recognition, we propose to compute fine grained handcrafted features extracted from GPS coordinates (e.g., number of visited places), which have been studied in [10]. The details about mobility features and description are shown in Table 1. Similarly, social interaction (e.g., number of encountered Bluetooth devices) and physical activity features (e.g., entropy of physical activity) are also extracted and shown in Table 1.

3.2 Graph-based Feature Engineering

Although handcrafted feature engineering can produce fine-grained and more interpretable features, they are designed using domain knowledge and intuition, which usually do not generalize well to different populations [60]. In addition, handcrafted features may not capture dynamic characteristics of human mobility, social interactions and physical activities. For example, in daily place visit trajectories, the impact of transitioning from one place to the next cannot be practically encoded by using handcrafted feature engineering given heterogeneous number of places visited by each person. To address this limitation, we represent daily GPS trajectory, Bluetooth encounters, and Physical activities as graphs, as shown in Fig 1 and apply GNNs to automatically generate deep features for our influenza-like symptoms recognition task.

We use $\mathcal{G}_p = (V_p, E_p, \mathbf{X_p})$ to represent daily GPS trajectory, where $V_p = \{v_{p_1}, v_{p_2}...\}$ and $E_p = \{e_{p_1}, e_{p_2}...|e_{p_i} \in V_p \times V_p\}$ are the set of nodes (places visited) and edges (transitions between places), respectively; and $\mathbf{X_p} = \{x_{p_1}, x_{p_2}, ... |x_{p_i} \in \mathbb{R}^d\}$ are the feature vectors associated with each node in $\mathcal{G}_p$, where d is the node features dimension. For Bluetooth encounter network, we represent each detected Bluetooth device as a node in the graph and use edges to indicate the adjacency between two detected Bluetooth devices. We use $\mathcal{G}_b = (V_b, E_b, \mathbf{X_b})$ to denote daily Bluetooth encounters , where $V_b = \{v_{b_1}, v_{b_2}...\}$ and $E_b = \{e_{b_1}, e_{b_2}...|e_{b_i} \in V_b \times V_b\}$ are the set of nodes (scanned devices) and edges (adjacency), respectively; and $\mathbf{X_b} = \{x_{b_1}, x_{b_2}, ...|x_{b_i} \in \mathbb{R}^d\}$ are the feature vectors associated

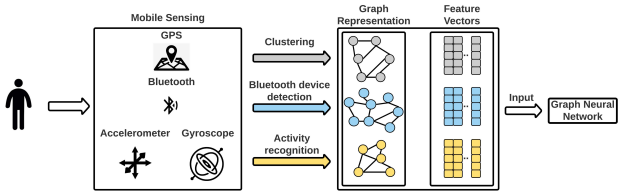

Figure 1: Graph Based Behavior Modeling using Mobile Sensing.

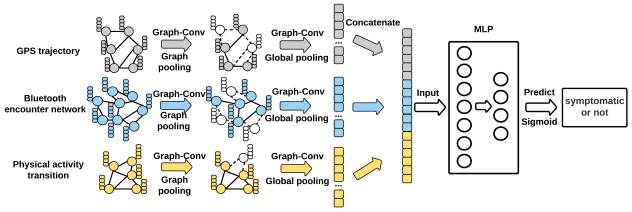

Figure 2: Multi-Channel Graph Neural Network for Influenza-like Symptom Recognition.

with each node in $\mathcal{G}_b$, where d is the node features dimension. To represent physical activity transitions, we encode each activity label as node in the transition graph denoted as $\mathcal{G}_a = (V_a, E_a, \mathbf{X_a})$, where $V_a = \{v_{a_1}, v_{a_2} ...\}$ and $E_a = \{e_{a_1}, e_{a_2} ... | e_{a_i} \in V_a \times V_a\}$ are the set of nodes (physical activities) and edges (transitions between physical activities), respectively; and $\mathbf{X_a} = \{x_{a_1}, x_{a_2}, ... | x_{a_i} \in \mathbb{R}^d\}$ are the feature vectors associated with each node in $\mathcal{G}_a$, where d is the node features dimension. All the graphs $\mathcal{G}_p, \mathcal{G}_b$, and $\mathcal{G}_a$ are unweighted and non-directional. As shown in Fig 1, behavior represented graphs and their corresponding feature vectors can be fed into graph neural networks, discussed in section 3.3, to build predictive models.

3.3 Graph Neural Networks and Node Embedding

Given the graphs representing daily mobility patterns, social interactions, and physical activities, we apply graph neural networks (GNNs) to automatically extract high level topological features from non-euclidean spaces in an optimized way. Specifically, GNNs can capture high order interaction information between neighbors in graphs and aggregate local node features to generate graph level numerical representation [54]. In our study, this prediction task can be formed as graph classification given multi-channel input of graph structured data, $\mathcal{G}_p, \mathcal{G}_b$ and $\mathcal{G}_a$. As shown in Fig 2, we propose an end-to-end Graph neural network architecture with multi-channel input to predict existence of influenza-like symptoms. The feature vectors of node in each graph can also be automatically learned by using Node Embedding, which maps each node into numerical space such that nodes who share similar topological structures will be closed with each other in the embedding numerical space. The details of node embedding selection will be demonstrated in section 5.2. We investigated four different state of the art node embedding techniques (Node2Vec, Walklets, NodeSketch, and BoostNE), and four different graph convolutional layers (GCN, GaphSAGE, GAT, GIN), as defined in the following subsections, to compare and contrast their performance in our symptom recognition task.

3.3.1 Graph Convolutional Layers **Graph Convolutional Networks (GCNs)** [27] are neural networks that perform graph convolution operations on graph structured inputs. GCNs can infer node level embeddings based on the features of node neighborhoods. Given a graph $\mathcal{G}(V, E, \mathbf{X})$, we denote the adjacency matrix and degree matrix of $\mathcal{G}$ as A and D respectively. Then layer-wise propagation in a GCN can be expressed as

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}). \quad (1)$$

We set each node in $\mathcal{G}$ as self-connected, and define $\tilde{A} = A + I$, where I is the identity matrix. $\tilde{D} = D + I$, and $W^{(l)}$ is the weight matrix which will be learned in the training process. l indicates the lth layer of graph convolution operations. $H^{(l)}$ is the matrix output with activation from lth graph convolution layer, and $H^{(0)} = \mathbf{X}$. $\sigma(\bullet)$ is the activation function, such as $ReLU(x) = max(0, x)$. With one layer of graph convolution, GCNs aggregate node features from the nodes' first-order neighborhood. Higher order neighborhood aggregation can be achieved by adding more convolution layers.

GraphSAGEs [24] utilize various aggregate functions to integrate features of node neighborhoods in search depth. Different from GCNs, which aggregate nodes' first order neighborhood information, GraphSAGEs apply forward propagation to concatenate the features of each local neighborhood through the propagation trace, such that higher order topological property can be leveraged to aggregated node features. The GraphSAGE convolution can be expressed as

$$h_v^{(l+1)} = \sigma(W^{(l)} \bullet CONCAT[h_v^{(l)}, F^{(l)}(\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\}]) \quad (2)$$

for each node v in V, where $CONCAT$ is the concatenation operation and $F^{(l)}(\bullet)$ is the aggregation function in layer l, which can be mean, max, sum, and LSTM [24]. $\mathcal{N}(\bullet)$ represents the set of neighborhoods of the node v.

Graph Attention Networks (GATs) [48] exploit attention mechanisms to produce node representations. Given sequential observations such as text data, attentions are usually realized by allocating more weights to the most important features from the inputs, such that more attentions will be focused on the important features when prediction is made. GATs adapt attention mechanisms by learning the weights for the node neighborhoods, and during the aggregation process, more weights will be assigned to the node neighborhoods that have more influence to the node. The graph attention convolution can be defined as

$$h_v^{(l+1)} = \sigma(\sum_{u \in \mathcal{N}_{(v)}} \alpha_{v,u}^{(l)} W^{(l)} h_u^{(l)}), \quad (3)$$

where $h_v^{(0)} = x_v$, and $\alpha_{v,u}^{(l)}$ is the attention of node v in the lth layer, which can be further expressed as

$$\alpha_{v,u}^{(l)} = \frac{exp(\eta(\mathbf{a}^{(l)T}[W^{(l)} h_v^{(l)}, W^{(l)} h_u^{(l)}])}{\sum_{u' \in \mathcal{N}(v)} exp(\eta(\mathbf{a}^{(l)T}[W^{(l)} h_v^{(l)}, W^{(l)} h_{u'}^{(l)}])}. \quad (4)$$

Here η is LeakyReLU activation function, while $\mathbf{a}$ is a weight vector to parametrize the attention mechanism.

Graph Isomorphic Networks (GINs) [56] generalize Weisfeiler-Lehman graph isomorphism test to better discriminate graphs. GINs

apply multi-layer perceptrons (MLP) to approximate the composition function as shown below:

$$h_v^{(l+1)} = MLP^{(l)}((1+\epsilon^{(l)}) \bullet h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} h_u^{(l)}), \quad (5)$$

where $\epsilon^{(l)}$ is a learnable parameter. GINs are argued as powerful as the Weisfeiler-Lehman graph isomorphism test for graph classification tasks [56].

3.3.2 Node Embedding Node embeddings are applied to infer node attributes in graphs, mapping graph structured data into numerical spaces. Node embeddings can automatically extract high level node features by leveraging dependencies between nodes and graph structures [25]. In general, there are two categories of Node embedding methods: 1) Deep walk [41]; 2) matrix factorization [35]. We applied two node embedding methods from each category listed below:

(1) **DeepWalk**
   (a) *Node2Vec* [23] deploys exploration-exploitation strategy to consider both homophily and structural equivalence in graphs when performs node embedding. By making depth first search and breadth first search for sampling the node neighborhoods, Node2Vec can generate continuous feature representations to encode the immediate local structure of nodes and global network neighborhoods.
   (b) *Walklets* [42] samples node neighborhoods context with skipping over nodes in each random walk. Multi-scale relationships can be encoded by subsampling fixed length of path in the node neighborhoods context generated from skipping random walks.
(2) **Matrix factorization**
   (a) *NodeSketch* [59] starts lower-order node embedding with generating Self-Loop-Augmented (SLA) adjacency matrix and then using hashing functions to map each SLA adjacency vector into Hamming space to approximate the similarity of each adjacency vector.
   (b) *BoostNE* [32] conducts a sequence of non-negative matrix factorization to the residual of the connectivity matrix approximated from previous step to generate sequence of weak embeddings. Then the sequence of weak embeddings will be ensembled to generate fine-grained embedding representations.

4 Data Collection and Description

The data used in this paper is a subset of a multicohort study that studied mobile sensing techniques for earlier diagnosis of illness. In total, 2700 participants distributed in 24 states in the U.S. were recruited to participate in this study for up to 1-year. We selected only participants that had at least 14 days worth of data from GPS, Bluetooth, Activity, and self-reported influenza-like symptom. Finally, Data from 448 participants are retained for our current work, and the active data collection date is from Feb 15th 2019 to Apr 30th 2020. The mean age of the participants is 40.71(sd = 11.52). 65.36% of the participants are female, with White being 66.2%, African American 19.3%, Asian 7.3%, multiple races 3.3%, Hispanic 3.5%, and others 0.4%. Among the 448 participants, 52% are full time workers, 20% are full time students, 12% are working part time, 7% have retired, 5% are care takers, and 4% are temporarily unemployed. More details about the larger study can be provided by the corresponding author.

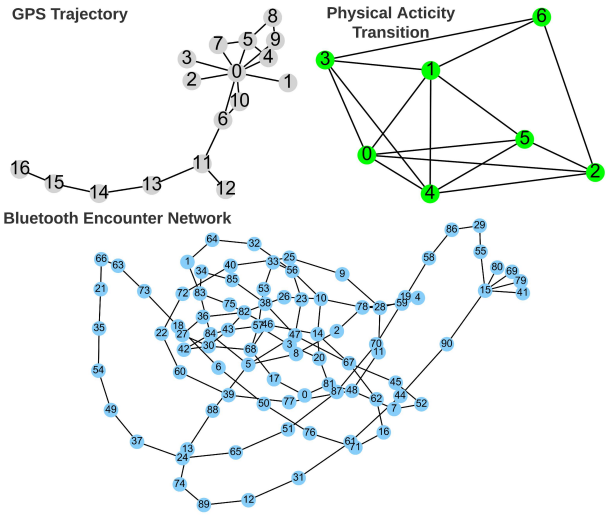

Figure 3: Samples of daily location trajectory, Bluetooth encounter network, and physical activity transitions graphs.

In the data collection process, participants were asked to install and run ReadiSens, a mobile sensing app, for up to 4 months. ReadiSens is a cross-platform app built on top of Sensus [55] to passively collect GPS location data every 30 mins; Bluetooth Encounters data every 15 mins in Android, and every 30 mins in iOS; activity data when activities change in Android, and every 2 hours in iOS. Ecological momentary assessments (EMAs) were delivered at 8pm everyday to collect self-reported symptoms, which include fever, cough, difficulty breathing, fatigue, muscle aches, headache, sore throat, runny nose, nausea, and diarrhea. Data were periodically uploaded to Amazon Web Services (AWS) Simple Storage Service (S3). All data were encrypted and anonymized to protect participants' data security and privacy. Any identifying information was omitted before being stored on AWS. GPS data were anonymized by omitting the integer parts of their longitudes and latitudes. Bluetooth's name and MAC addresses were automatically hashed on Readisense.

5 Experiments and Results

In this section, we illustrate the detailed information about data preprocessing, graph modeling, GNN implementation, model selection, interpretability of GNN and hyperparameter sensitivity analysis.

5.1 Data Preprocessing and Statistics

After data preprocessing, we are left with 8,415 observations, among which 6,923 are non-symptomatic (no influenza-like symptoms reported) samples , and 1,492 (17.7% of all samples) are symptomatic (at least one influenza-like symptoms reported) samples. Given the discrete signals from daily GPS trajectories, Bluetooth encounters and activity transitions traces, we convert them into graph representations. Figure 3 provided one sample of the constructed graphs

Symptom	Node Number			Average Node Degree			Connectivity			Assortativity		
	GPS	Bluetooth	Activity	GPS	Bluetooth	Activity	GPS	Bluetooth	Activity	GPS	Bluetooth	Activity
NO	6.32	180.70	5.02	3.25	2.79	4.96	1.08	1.03	2.87	-0.33	-0.06	-0.10
YES	6.00	157.45	4.86	2.91	2.74	4.82	1.07	1.02	2.86	-0.39	-0.14	-0.11

Table 2: Compare the average graph metrics of GPS trajectory, Bluetooth encounter networks, and Physical activity transitions between days of participants with influenza-like symptoms and those without any symptoms.

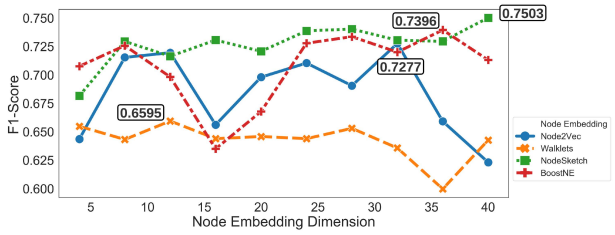

Figure 4: F1-score comparison for the combinations of Node Embedding methods and embedding dimensions. GNNs with two GCN layers were built with TopK (ratio = 0.75) graph pooling layer and global sum pooling layer. The training processes were set using 40 epochs, 64 batch size, 10 step patience for early stopping, and learning rate 0.001 for Adam optimizer.

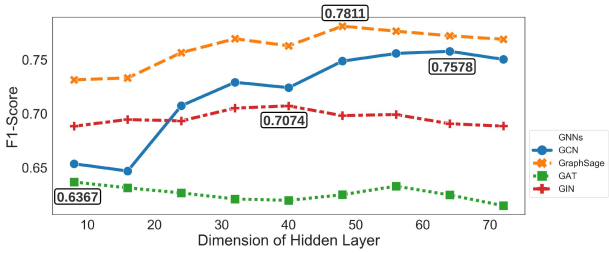

Figure 5: F1-score comparison for the dimensions of neurons for the hidden layers in GNNs. GNNs consisted of two Graph convolutional layers with TopK (ratio = 0.75) graph pooling layer and global sum pooling layer using the entire dataset. The training processes were set using 80 epochs, 256 batch size, 15 step patience for early stopping and learning rate = 0.001 for Adam optimizer.

for each sensing modality. To illustrate and compare graph properties for robust and symptomatic samples, we present summary statistics of graph metrics in Table 2. In the symptomatic cases, the average node number, node degree, connectivity and assortativity are consistently lower than those in the non-symptomatic cases. These discrepancies indicate that the graphs in symptomatic cases generally have less vertices and less edges, implying that the participants, who have influenza-like symptoms, could have less mobility, social interactions, and diverse physical activities. Since we had imbalanced distribution between symptomatic and non-symptomatic cases, we oversampled the minor cases in our experiments. For a fair comparison between different GNNs [16], we separated the whole dataset into 90% training data and 10% testing data, and in the 90% training data, we select 10% training data as validation data and applied oversampling technique to re-balance the rest of the training data. In the end, we had 10,239 training samples after oversampling, 692 validation samples, and 841 testing samples. To deal with missing graphs in the data set, we imputed naive graphs which had 2 nodes with one edge connecting them.

5.2 Node Embedding Selection

To select the best node embedding technique with sufficient embedding dimensions, we experimented the above mentioned node embedding methods using GCNs with increasing embedding dimensions for one run. We used two layers of GCNs with TopK graph pooling and global sum pooling layer for 20% random sampled data. F1-scores from these experiments are reported in Figure 4. Matrix approximation methods (NodeSketch, BoostNE), showed better performance than DeepWalk methods (Node2Vec, Walklets). This could be because DeepWalk methods usually require large number of sampled node neighborhoods to generate the contexts of the embedded nodes. However, the average node degrees in our data are generally small such that sampled random walks could be insufficient to learn good latent representations. We select NodeSketch with an embedding dimension of 40 based on its performance.

5.3 Hidden Layer Dimension Selection

The prediction performance of deep learning methods can be impacted by dimension of neurons in the hidden layers. Neural networks with high dimensional layers generally can have more predictive power, but require larger training samples. In this section, we tuned the dimension of graph convolutional layers to select the number of dimensions for each GNN. We set up GCN, GraphSAGE, GAT and GIN with the same TopK graph pooling layer and global sum pooling layer, and trained the GNNs with the full dataset with one run. NodeSkech were implemented to extract node embeddings. F1-scores were reported Figure 5. We observed that the F1-scores generally increased with increasing dimension of hidden layers. However GAT and GIN were much less sensitive to the variation of hidden layer dimensions. We selected dimensions 64 for GCN, 48 for GraphSAGE, 8 for GAT, and 40 for GIN.

5.4 Pooling layer selection

In this section, we fine-tuned the GNNs with different combinations of graph pooling layers and global pooling layers, as shown in Figure 3.3. Graph pooling layer, including TopK pooling [14] and SAG Pooling [31] in this study, is to select most important nodes (generating smaller graphs) with respect to the prediction performance. TopK pooling selects the top K nodes in graphs by ranking the projected scalar values of each node features. SAG

Graph Convolutional Layer	Graph and Global Pooling Layer					
	TopK Pooling			SAG Pooling		
	GlobalSum	GlobalMax	GlobalAttention	GlobalSum	GlobalMax	GlobalAttention
GCN	0.7578	**0.8617**	0.7551	0.7881	0.8591	0.8081
GraphSAGE	0.7811	**0.8856**	0.7905	0.8068	0.8792	0.7524
GAT	0.6367	0.6309	0.6072	0.5882	**0.6409**	0.6206
GIN	0.7072	0.7313	0.6925	0.7243	**0.7441**	0.7244

Table 3: F1-score comparison to select best combination of graph pooling layer and global pooling layer in Figure 2 in GNNs. The dimensions of each GNN were set up based on the results from section 5.3. TopK pooling ratio = 0.75. The training processes were set with 80 epochs, 256 batch size, 15 step patience for early stopping and learning rate = 0.001 for Adam.

pooling applies attention mechanism to select top ranked nodes by self-attention scores. Global pooling layer (read-out layer), including GlobalSum, GlobalMax, and GlobalAttention [33] in this study, is to generate graph level features from node level features. GlobalSum and GlobalMax produce graph level representations by taking sum and max of each node feature in the feature vectors, respectively, while GlobalAttention uses neural networks to create self-attentioned graph level representation [33]. The results compared by F1-score were shown in Table 3. We observed that in both TopK and SAG poolings, GlobalMax pooling produced the best results. This could be because GlobalMax can extract the most salient features from node feature vector to better represent the graphs.

5.5 Results and Analysis

To compare handcrafted and graph based behavior modeling, we select the following baseline models for handcrafted features: 1) Logistic regression [40], 2) Support vector machine (SVM) [45], 3) Random forest [34], 4) Xgboost [15], 5) Multi Layer Perceptron (MLP) [19], all of which have been widely applied and investigated in mobile sensing studies[28, 29, 36]. Equivalently, we replaced missing values with 0 in handcrafted features, and applied oversampling to re-balance the training data set. We set up the GNNs with the best performed node embedding method, their own selected hidden layer dimensions, and pooling layer combinations as discussed in section 5.2, 5.3 and 5.4. The hyperparameter setting is as follow: TopK pooling ratio = 0.75, epochs = 80, batch size = 256, early stopping patience = 15 and learning rate = 0.001. We performed 10-fold cross validation to compare the performance between handcrafted and graph based methods. The metrics for comparison include precision, recall, F1-score, and AUC-score.

The results are shown in Table 4. We observe that except GAT, all other GNNs outperformed the baseline models with handcrafted feature engineering by all evaluation metrics. From the training and validation loss plot generated from one cross validation iteration (Figure 6), the GCN training loss continued to decrease, but the validation loss reached a minimum between epoch 50 and 60, implying that GCN could have been overfitted in the training data set after epoch 60. For GIN, the training loss decreased smoothly, but the validation loss did not show significant improvement after 80 epochs. This could be because GIN had much more parameters to tune, since GIN used MLPs as the composition function. For GAT, both the training and validation loss did not show any improvements, thus early stopping was activated at 68 training epoch.

Category	Model	Metrics			
		Precision	Recall	F1	AUC
Baseline	Logistic Regression	0.5673 ± 0.0096	0.6139± 0.0171	0.5896 ± 0.0129	0.6532 ± 0.0208
	SVM	0.5823 ± 0.0102	0.6414± 0.0177	0.6104 ± 0.0135	0.6838 ± 0.0133
	Random Forest	0.6617 ± 0.0308	0.6252±0.0222	0.6429 ± 0.0257	0.7339 ± 0.0151
	Xgboost	0.6047 ± 0.0111	0.6610± 0.0175	0.6316 ± 0.0137	0.7354 ± 0.0156
	MLP	0.6407 ± 0.0212	0.6679± 0.0198	0.6501 ± 0.0186	0.7443 ± 0.0162
GNN	GCN	0.8079 ± 0.0246	0.8549 ± 0.02446	0.8307 ± 0.0236	0.9148 ± 0.0195
	GraphSAGE	**0.8546 ± 0.0210**	**0.8991 ± 0.0164**	**0.8762 ± 0.0184**	**0.9539 ± 0.0132**
	GAT	0.6087 ± 0.0065	0.6872 ± 0.0133	0.6455 ± 0.0058	0.6725 ± 0.0091
	GIN	0.7176 ± 0.0153	0.7733 ± 0.0223	0.7443 ± 0.0161	0.8179 ± 0.0192

Table 4: Results comparison between baseline models with handcrafted feature engineering and GNNs

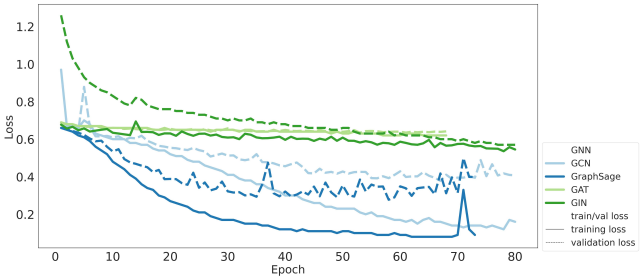

Figure 6: Training and validation loss comparison of GNNs over epochs.

GraphSAGE provided the best performance among the GNNs as it produced lower training and validation loss than the other GNNs in almost all training epochs. GraphSAGE showed the best predictive power because it can aggregate higher order node neighborhood information such that more complex interaction between nodes can be leveraged to classify the input graphs. Additionally, since the input graphs generally have large number of nodes but small number of node degree, the node neighbors' topological structures can be better represented in GraphSAGE, whereas in GCN only local node structures could be aggregated.

5.6 Interpretability of Graph Neural Networks

To build trust in the black box GNN approaches, and increase the transparency of GNN's decision making mechanism, we applied Class Activation Mapping (CAM) [61] to interpret contributions of the node features and graph structures to GraphSAGE's prediction. CAMs have been widely used in image recognition to explain how high level image features can be used in convolutional neural networks for classification tasks[44]. In this study, we implement the GNN version of CAM to identify important node features in differentiating graph classes. GNN based CAM can analyze the output of last graph convolutional layer and link the learned feature weights with input node attribute to calculate node importance for graph classification. We adapt our GraphSAGE model for a better interpretation by replacing global max pooling layer with global average pooling layer in the last graph convolutional layer and remove the dense layers as discussed in [61]. We generate the heat maps of nodes in graphs from both symptomatic and non-symptomatic cases, as shown in Figure 7. In general, we observe that the graphs of non-symptomatic cases are more diverse and complex than symptomatic cases. The graphs also show that that higher number of visited places, diversity of locations, and physical

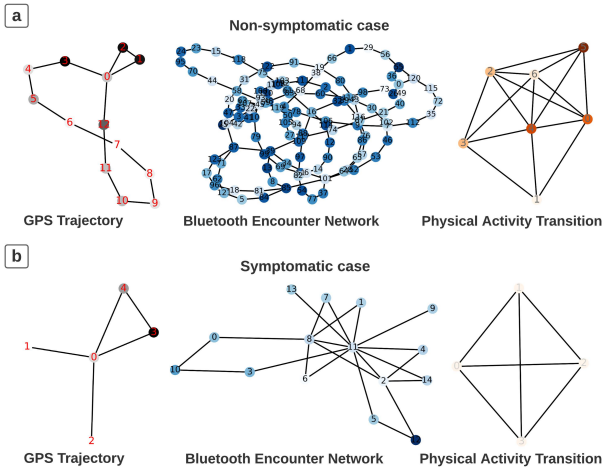

Figure 7: Interpretable GraphSAGE: (a) is a sampled Non-symptomatic case, and (b) is a sampled Symptomatic case. The darker the nodes the more important of this node will be to predict existence of influenza-like symptoms. The numbers shown in each node indicate the order of being in each node.

activity are all indicative of the absence of symptoms. This can indicate that our model was able to learn that when users show a lower activity levels and a more sedentary behavior, they have higher chances of having ILI symptoms.

We also observed that nodes from different cases, different graphs, and different sub-graph structures can have different importance levels at predicting symptoms. In non-symptomatic case, as shown in 7 (a), important nodes were distributed globally in multiple sub-structures in each graph. This implies that multiple nodes in each non-symptomatic graph have high contribution to the symptom recognition. In symptomatic case, as shown in 7 (b), important nodes were distributed locally in limited sub-graphs. This can speak to the importance of some contextual situations (e.g., home and hospitals) at differentiating between symptomatic and non-symptomatic cases.

5.7 Hyperparameter Sensitivity Analysis

For the fine-tuned GraphSAGE model, we investigated several critical hyperparameters that could impact the prediction performance. The results were shown in Figure 8. In plot (a), we found that higher learning rate produces worse prediction performance, and also generates larger variance in validation loss, which means the model has larger variance in its predictions. In plot (b), different TopK ratios lead to different F1-scores ranging from 0.87 to 0.91. There is no clear pattern on how the preference of TopK ratios can have impacts on prediction performances, but from the experiments a 0.6 TopK ratio returns the best F1-score at 0.91. This implied that top 60% of nodes had higher contribution to predict existence of influenza-like symptoms than other nodes. In plot (c), we observed that with only 50% of data as the training set, GraphSAGE could have comparable validation predictive performance as the models trained with more than 50% data. This has practical implication

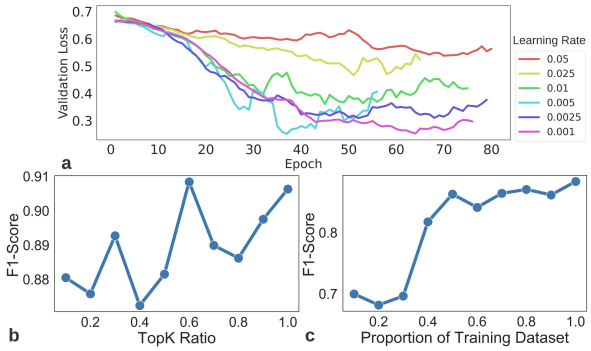

Figure 8: Hyperparameter sensitivity analysis for GraphSAGE: a) learning rates on validation loss over training epochs; b) prediction performances on different TopK ratios; c) prediction performances using different proportions of data as training set.

as in mobile sensing and human centered computing, samples are expensive to obtain.

6 Conclusion, Discussions, and Future Works

In this study, we applied graphs to represent symptomatic human behaviors captured by multi-modal mobile sensors, and demonstrated the potential of graph neural networks to detect and predict human states. Based on our best knowledge, this is the first work of graph-based behavior modeling by leveraging graph neural networks and multi-modal mobile sensing. Instead of using handcrafted feature engineering to generate fine-grained behavior markers, we proposed to use graph representations to encode the dynamics of human behavior state transitions, and node embedding techniques to extract topological node attributes, and finally GNNs to learn automatic deep features for influenza-like symptom recognition. We showed that GNNs with GraphSAGE convolutional layers significantly outperform baseline models with handcrafted features. Additionally, the graph-based behavior modeling and mobile sensing framework, as shown in Figure 1 can be generalized to applications in mobile sensing, such as those in mental health [52].

In addition to the generalization of this framework in the mobile sensing field, this work also has public health application with regard to overcoming the limitations of traditional public health reporting systems and syndromic surveillance. On an individual level, influenza-like symptoms can be detected passively and unobtrusively by using this framework, and interventions can be delivered to change individuals' behaviors such as self-isolation. On a community and population level, early detection of influenza-like symptoms can inform people of the potential of being infected by those who are sick and in proximity, and help predict nationwide influenza levels for public health policy decisions.

Our current work is subject to the following limitations. First of all, we built global models using data from all participants by assuming homogeneity of behaviors and personalities across different participants in our study. A better approach would be to create personalized models by retraining the global model with individual data. Unfortunately, our dataset was not adequate for building personalized model due to missing data and the relatively short study

period. Secondly, using Bluetooth encounters as proximity of real face-to-face social interactions could be biased. This approximation is based on the assumption that more detected Bluetooth encounters lead to higher probability of face-to-face social interactions [20], which is not always the case. Thirdly, lab-confirmed diagnosis of influenza is not provided. Influenza symptomatology is nonspecific and similar to many other diseases at an early stage which limits the application of this work. Last, due to privacy concerns and risks of identity exposures, we anonymized the GPS coordinates and encrypted Bluetooth encounter device IDs. Because of this, important semantic and contextual information of the behavior states represented by our graphs is lost, which greatly weakens the interpretability of the decision process.

In the future, we will extend our current work in two directions. First, by extending the sequence of observations of mobile sensing data to several days, dynamic graphs through time line can be used to capture temporal variations of behavior states, such that spatiotemporal representations of human behaviors can be extracted to analyze dynamic interactions between symptoms of diseases and human behaviors over time. Second, we plan to use semi-supervised machine learning techniques to leverage both labeled and unlabeled observations to produce models with better generalization.

Acknowledgement

This work was supported by the DARPA Warfighter Analytics using Smartphones for Health (WASH) program.

References

[1] Forsad Al Hossain, Andrew A Lover, George A Corey, Nicholas G Reich, and Tauhidur Rahman. 2020. FluSense: a contactless syndromic surveillance platform for influenza-like illness in hospital waiting areas. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–28.

[2] Mawulolo K Ameko, Lihua Cai, Mehdi Boukhechba, Alexander Daros, Philip I Chow, Bethany A Teachman, Matthew S Gerber, and Laura E Barnes. 2018. Cluster-based approach to improve affect recognition from passively sensed data. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 434–437.

[3] Apple. 2020. *CMMotionActivity*. https://developer.apple.com/documentation/coremotion/cmmotionactivity

[4] Gianni Barlacchi, Christos Perentis, Abhinav Mehrotra, Mirco Musolesi, and Bruno Lepri. 2017. Are you getting sick? Predicting influenza-like symptoms using human mobility behaviors. *EPJ Data Science* 6, 1 (2017), 27.

[5] David Blumenthal, Elizabeth J Fowler, Melinda Abrams, and Sara R Collins. 2020. Covid-19—Implications for the Health Care System.

[6] Mehdi Boukhechba, Anna N Baglione, and Laura E Barnes. 2020. Leveraging Mobile Sensing and Machine Learning for Personalized Mental Health Care. *Ergonomics in Design* (2020), 1064804620920494.

[7] Mehdi Boukhechba and Laura E Barnes. 2020. Swear: Sensing using wearables. Generalized human crowdsensing on smartwatches. In *2019 IEEE 11th International Conference on Applied Human Factors and Ergonomics. IEEE*.

[8] Mehdi Boukhechba, Lihua Cai, Philip I Chow, Karl Fua, Matthew S Gerber, Bethany A Teachman, and Laura E Barnes. 2018. Contextual analysis to understand compliance with smartphone-based ecological momentary assessment. In *Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare*. 232–238.

[9] Mehdi Boukhechba, Lihua Cai, Congyu Wu, and Laura E Barnes. 2019. ActiPPG: using deep neural networks for activity recognition from wrist-worn photoplethysmography (PPG) sensors. *Smart Health* 14 (2019), 100082.

[10] Mehdi Boukhechba, Alexander R Daros, Karl Fua, Philip I Chow, Bethany A Teachman, and Laura E Barnes. 2018. DemonicSalmon: Monitoring mental health and social interactions of college students using smartphones. *Smart Health* 9 (2018), 192–203.

[11] Mehdi Boukhechba, Yu Huang, Philip Chow, Karl Fua, Bethany A Teachman, and Laura E Barnes. 2017. Monitoring social anxiety from mobility and communication patterns. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. 749–753.

[12] Lihua Cai, Mehdi Boukhechba, Matthew S Gerber, Laura E Barnes, Shayna L Showalter, Wendy F Cohn, and Philip I Chow. 2020. An integrated framework for using mobile sensing to understand response to mobile interventions among breast cancer patients. *Smart Health* 15 (2020), 100086.

[13] Lihua Cai, Mehdi Boukhechba, Congyu Wu, Philip I Chow, Bethany A Teachman, Laura E Barnes, and Matthew S Gerber. 2018. State affect recognition using smartphone sensing data. In *Proceedings of the 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*. 120–125.

[14] Cătălina Cangea, Petar Veličković, Nikola Jovanović, Thomas Kipf, and Pietro Liò. 2018. Towards sparse hierarchical graph classifiers. *arXiv preprint arXiv:1811.01287* (2018).

[15] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[16] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. 2019. A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893* (2019).

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[18] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 417–426. https://doi.org/10.1145/3308558.3313488

[19] Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.

[20] Michele Girolami, Fabio Mavilia, and Franca Delmastro. 2020. Sensing social interactions through ble beacons and commercial mobile devices. *Pervasive and Mobile Computing* 67 (2020), 101198.

[21] Jiaqi Gong, Yu Huang, Philip I Chow, Karl Fua, Matthew S Gerber, Bethany A Teachman, and Laura E Barnes. 2019. Understanding behavioral dynamics of social anxiety among college students through smartphone sensors. *Information Fusion* 49 (2019), 57–68.

[22] Google. 2020. *Google Activity Recognition API*. https://developers.google.com/location-context/activity-recognition

[23] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[24] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[25] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[26] Hongping Hu, Haiyan Wang, Feng Wang, Daniel Langley, Adrian Avram, and Maoxing Liu. 2018. Prediction of influenza-like illness based on the improved artificial tree algorithm and artificial neural network. *Scientific reports* 8, 1 (2018), 1–8.

[27] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[28] Nicholas D Lane and Petko Georgiev. 2015. Can deep learning revolutionize mobile sensing?. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. 117–122.

[29] Francisco Laport-López, Emilio Serrano, Javier Bajo, and Andrew T Campbell. 2020. A review of mobile sensing systems, applications, and opportunities. *Knowledge and Information Systems* 62, 1 (2020), 145–174.

[30] Oscar D Lara and Miguel A Labrador. 2012. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials* 15, 3 (2012), 1192–1209.

[31] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. *arXiv preprint arXiv:1904.08082* (2019).

[32] Jundong Li, Liang Wu, Ruocheng Guo, Chenghao Liu, and Huan Liu. 2019. Multi-level network embedding with boosted low-rank matrix approximation. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 49–56.

[33] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).

[34] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.

[35] Xin Liu, Tsuyoshi Murata, Kyoung-Sook Kim, Chatchawan Kotarasu, and Chenyi Zhuang. 2019. A general view for network embedding as matrix factorization. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 375–383.

[36] Zheng Lou, Lili Wang, and Guozhen Shen. 2018. Recent advances in smart wearable sensing systems. *Advanced Materials Technologies* 3, 12 (2018), 1800444.

[37] Fenglong Ma, Shiran Zhong, Jing Gao, and Ling Bian. 2019. Influenza-Like Symptom Prediction by Analyzing Self-Reported Health Status and Human

Mobility Behaviors. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 233–242.

[38] Daniel Neil, Joss Briody, Alix Lacoste, Aaron Sim, Paidi Creed, and Amir Saffari. 2018. Interpretable graph convolutional neural networks for inference on noisy knowledge graphs. *arXiv preprint arXiv:1812.00279* (2018).

[39] Donald R Olson, Kevin J Konty, Marc Paladini, Cecile Viboud, and Lone Simonsen. 2013. Reassessing Google Flu Trends data for detection of seasonal and pandemic influenza: a comparative epidemiological study at three geographic scales. *PLoS Comput Biol* 9, 10 (2013), e1003256.

[40] Fred C Pampel. 2020. *Logistic regression: A primer*. Vol. 132. SAGE Publications, Incorporated.

[41] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. *CoRR* abs/1403.6652 (2014). arXiv:1403.6652 http://arxiv.org/abs/1403.6652

[42] Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. 2016. Walklets: Multi-scale graph embeddings for interpretable network classification. *arXiv preprint arXiv:1605.02115* (2016).

[43] Piero Poletti, Roberto Visintainer, Bruno Lepri, and Stefano Merler. 2017. The interplay between individual social behavior and clinical symptoms in small clustered groups. *BMC infectious diseases* 17, 1 (2017), 521.

[44] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. 2019. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10772–10781.

[45] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9, 3 (1999), 293–300.

[46] Wen Torng and Russ B Altman. 2019. Graph convolutional neural networks for predicting drug-target interactions. *Journal of Chemical Information and Modeling* 59, 10 (2019), 4131–4149.

[47] Kristin van Barneveld, Michael Quinlan, Peter Kriesler, Anne Junor, Fran Baum, Anis Chowdhury, Pramod N Junankar, Stephen Clibborn, Frances Flanagan, Chris F Wright, et al. 2020. The COVID-19 pandemic: Lessons on building more equal and sustainable societies. *The Economic and Labour Relations Review* 31, 2 (2020), 133–157.

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[49] Svitlana Volkova, Ellyn Ayton, Katherine Porterfield, and Courtney D Corley. 2017. Forecasting influenza-like illness dynamics for military populations using neural networks and social media. *PloS one* 12, 12 (2017), e0188941.

[50] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic Flow Prediction via Spatial Temporal Graph Neural Network. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 1082–1092. https://doi.org/10.1145/3366423.3380186

[51] Congyu Wu, Mehdi Boukhechba, Lihua Cai, Laura E Barnes, and Matthew S Gerber. 2018. Improving momentary stress measurement and prediction with bluetooth encounter networks. *Smart Health* 9 (2018), 219–231.

[52] Congyu Wu, Lihua Cai, Matthew S Gerber, Mehdi Boukhechba, and Laura E Barnes. 2018. Vector Space Representation of Bluetooth Encounters for Mental Health Inference. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 1691–1699.

[53] Ning Wu, Xin Wayne Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning Effective Road Network Representation with Hierarchical Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 6–14. https://doi.org/10.1145/3394486.3403043

[54] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[55] Haoyi Xiong, Yu Huang, Laura E. Barnes, and Matthew S. Gerber. 2016. Sensus: A Cross-Platform, General-Purpose System for Mobile Crowdsensing in Human-Subject Studies. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 415–426. https://doi.org/10.1145/2971648.2971711

[56] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[57] Mohammad Yamin. 2020. Counting the cost of COVID-19. *International Journal of Information Technology* (2020), 1–7.

[58] Chao-Tung Yang, Yuan-An Chen, Yu-Wei Chan, Chia-Lin Lee, Yu-Tse Tsan, Wei-Cheng Chan, and Po-Yu Liu. 2020. Influenza-like illness prediction using a long short-term memory deep learning model with multiple open data sources. *The Journal of Supercomputing* (2020), 1–27.

[59] Dingqi Yang, Paolo Rosso, Bin Li, and Philippe Cudre-Mauroux. 2019. Nodesketch: Highly-efficient graph embeddings via recursive sketching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1162–1172.

[60] Alice Zheng and Amanda Casari. 2018. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.".

[61] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.