Rapid Betweenness Centrality Estimates for Transportation Networks using Capsule Networks

Abdul Matin, Samuel Armstrong, Saptashwa Mitra, Shrideep Pallickara, Sangmi Lee Pallickara
Computer Science Department
Colorado State Unviersity
{Abdul.Matin, Sam.Armstrong, Saptashwa.Mitra, Shrideep.Pallickara, Sangmi.Palickara}@colostate.edu

***Abstract*—Measuring importance of nodes in a graph is one of the key aspects in graph analysis. Betweenness centrality (BC) measures the amount of influence that a node has over the flow of information in a graph. However, the computation complexity of calculating BC is extremely high with large-scale graphs. This is especially true when analyzing the road networks with millions of nodes and edges. In this study, we propose a deep learning architecture RoadCaps to estimate BC with sub-second latencies. RoadCaps aggregates features from neighbor nodes using Graph Convolutional Networks and estimates the node level BC by mapping low-level concept to high-level information using Capsule Networks. Our empirical benchmarks demonstrates that RoadCaps outperforms base models such as GCN and GCNFCL in both accuracy and robustness. On average, RoadCaps generates a node's BC value in 7.5 milliseconds.**

***Index Terms*—Betweeness Centrality, Machine learning, GCN, Capsule Network, Road Network Analysis**

I. Introduction

Natural disasters cause substantial disruptions to a community's transportation network, and natural phenomena extremes are predicted to increase both in their frequency and intensity [1]. When sections of roads are flooded, covered by debris, or suffer structural damage, the number of accessible roads and intersections are reduced, which contributed to a further reduction in the number of available routes. Temporary or permanent changes to road networks lead to unexpected traffic spikes over the remainder of the network. The consequences are even more severe for areas with limited number of routes (such as rural areas). An immediate and comprehensive understanding of the impact of infrastructure loss is critical for planning timely responses and evacuation strategies.

Traditionally, the impact of fast-evolving disrupted road networks and its complex influences have been analyzed with diverse methodologies such as stochastic optimization processes [2], demand, supply models and traffic analysis [3]. Recently, network analysis has been applied to road networks to analyze the influence of various natural disasters such as earthquakes and flooding [4].

Road networks can be represented as a planar graph that is a graph embedded in the plane with the graph's constituent edges representing physical road connections [5]. Road networks are distinguished from other networks, such as social networks. Since each vertex and edges are physically anchored to their geospatial locations and their network topology is relatively limited in terms of the number of long-range edges and number of edges associated with a single node [6]. Therefore, instead of degree-based metrics, metrics that can provide non-local, higher-level information such as network centralities have been widely adopted in road network analysis [7]. Betweenness centrality (BC) is one of the well-studied centrality measures, and has been used by several road network topography analyses based on betweenness centrality [8]. Betweenness centrality measures the importance of a link based on the amount of flow at a location. Betweenness centrality is a path-based measure calculated based on the number of shortest paths within a planar graph that passes through the vertex (e.g., intersection) [9]. However, the calculation of betweenness centrality measures over large-scale complex road systems in real-time poses critical computational challenges. First, computing betweenness centrality over highly complex large road networks is prohibitively expensive. Brandes' algorithm [10] for computating the betweenness centrality has a time complexity of $O(nm + n^2 logn)$ and the space complexity is $O(n + m)$, where n and m are the number of vertices and edges in a graph, respectively. With the complexity and abundant data of modern road networks (for e.g., the road system of in the state of California comprises more than 2.67 million intersections) computing the betweenness centrality measures in real-time is infeasible. Second, since betweenness centrality measures depend on the number of shortest paths flowing through the target location, they are easily influenced by the partial changes within the networks. Removing one edge may require recalculation of for a substantial area around the removed edge. Finally, for a large road network, the computation over a subarea may cause significant inaccuracies for nodes close to the boundary of an area. This *boundary effect*, in particular, introduces challenges for distributed approaches to calculation of betweenness centrality over a large spatial extent. In this study, we propose a deep learning-based approach, RoadCaps for road-vulnerability analysis calculating weighted BC measures with sub-second latencies over large and complex transportation networks. We combine aspects of Graph Neural Networks [11] and Capsule Networks to accomplish this. Topological characteristics and geospatial features of the surrounding area are extracted and factored into the model to achieve higher model generalization to support varying levels of complexity over the road network and locations of the target intersections. Sub-second inference latencies supported by our network are suitable for applications that need faster turnaround times.

A. Research Questions

In this paper, we explore the following research questions.

RQ-1: How can we estimate betweenness centrality accurately and rapidly at scale to support applications with interactive explorations of road importance while providing reliable accuracy? Achieving robust accuracy across topological locations is important to avoid boundary effects.

RQ-2: How can we incorporate geospatial characteristics at a given location with topological information to improve accuracy of the estimations?

RQ-3: How can a system estimate the betweenness centrality of a node with limited computing resources? Estimating betweenness centrality should not trigger calculating betweenness centrality for the entire road network. Also, each computation must be lightweight enough to be portable.

B. Approach Summary

In this study, we propose a deep network, RoadCaps, that estimates accurate betweenness centrality measures over complex road networks at sub-second latencies. RoadCaps captures nonlinear relationships between the weighted BC values and topological characteristics of the surrounding area combined with area-specific structural road characteristics. RoadCaps leverages capsules to capture hierarchical structural relationships between target intersection(s) and their proximate intersections. Capsule Neural Networks (CapsNets) have been successfully applied in computer vision and graph theory and demonstrably outperform traditional convolutional layers. To generate inputs to capsule layers that effectively snapshot topological and geospatial features, RoadCaps comprises multiple convolutional graph layers. Compared to existing GNNs that primarily target graph classification tasks [12], RoadCaps provides a novel regression capability that estimates 1 or more BC estimates for intersections. As part of this research, we constructed a topological graph representation of road networks and extracted highly relevant features. We introduced a feature for intersections, traffic tendency that encapsulates traffic capacity for a road segment. We have also designed a novel space-efficient data structure, GeoDensityMap, that tracks the complexity of a large road system. We have evaluated our methodology with a road network dataset for the state of California in the U.S. RoadCaps demonstrates a HUBER error of 2.054 on average, which represents a 31.08% improvement in accuracy compared to both model GCN and model GCNFCL. We performed a variogram analysis to evaluate RoadCaps's capability to address boundary effects that arise. RoadCaps demonstrated a consistently stable model performance across the state of California. On average, our model estimates single point BC in 7.5 milliseconds and 500 points BC in 24.26 milliseconds.

C. Paper Contributions

We have designed a scalable model that estimates the weighted betweenness centrality of intersections in a large road network. Our contributions include the following:

- Fast and accurate estimations of the weighted betweenness centrality of nodes in a large road network: Our model generates BC measures while accounting for the topological characteristics of proximate nodes and road network-specific features; crucially, this is performed at sub-second latencies.
- Highly generalizable estimations: Our model performance is robust to the topological variations of the road network.
- Wide applicability for other network centrality metrics: The proposed methodology is applicable for other network centrality metrics such as percolation centrality and eigenvector centrality.
- Light weight computing to accurately estimate betweenness centrality: Our system allows the users to estimate accurate BC without performing expensive computing tasks required in traditional BC calculations.

D. Paper Organization

Section 2 describes the background and related work. Our methodology is described in Section 3. Section 4 describes our empirical benchmarks alongside a discussion of the results. Section 5 describes related work. Finally, our conclusions and future work are described in Section 6.

II. Background and Dataset

A. Betweenness Centrality Analysis for Road Networks

Centrality analysis is widely used to measure node importance at local and global spatial scales. Local centrality is measured between nodes within a given radius while global centrality calculates the distance between nodes within a whole system. The centrality index is useful to understand the operational impact in terms of the network flow tendencies based on topological characteristics, e.g., airline networks, road networks, power networks, and canal networks. Frequently used metrics to estimate network centrality include: betweenness, closeness, straightness, and degree. Closeness centrality is a way of detecting the capability of nodes to spread information efficiently by means of measuring the inverse distance to all other nodes [13]. The straightness index considers the degree of straightness of the path to determine the effectiveness of the connectivity [14]. The degree of centrality is based on the count of the total number of connecting edges to a node [15].

In road network analysis, betweenness centrality analysis has been widely used due to its close correlation to global traffic flows within the network [16]. If two areas are connected by a small number of links, the removal of these links will disable the high volume of traffic flowing between the two areas. Therefore, measuring BC is one of the primary interests of road network resilience to natural disasters. As seen in (1),the betweenness centrality of a node (k) is the total number of shortest paths at node (k) divided by the total number of shortest paths that exist between two nodes (i and j) of a given radius (r).

$$Betweenness[k]^r = \sum_{i \neq j \neq k \epsilon d[i,j] \leq r_i}^{n} \frac{N_{d[i,j]}[K]}{N_{d[i,j]}} \tag{1}$$

Betweenness analysis has been applied to weighted graphs effectively. The first step of the estimation of betwenness centrality is the shortest path calculation. Shortest path between source and destination points is determined as the path with smallest total weight. The weight of the edge is inversely related to the travel time and the number of lanes of a road. So, any intersection points will have high betweenness centrality if more paths with smaller weights pass through it.

B. Dataset and Study Areas

In this study, we have used transportation datasets provided by OpenStreetMaps [17] for the state of California, U.S. California is the third largest state in the U.S. by area (163,696 square miles) with diverse geographical landscapes, mountains, beaches, lakes, and large city areas. The road network in California contains more than 2.67 million intersections with more than 4.45 million miles of state and county highways. Other types of roads are primary, secondary, tertiary, trunk, service, pedestrian, bike, race, residential, and so on. We selected highways, primary, secondary, tertiary, and trunk roads to focus on land transportation, particularly for major roads that are used by auto vehicles.

The dataset also provides the latitude and longitude information of each intersection point, length, maximum speed, number of lanes, and direction of the road. A graph was made from the extracted road network: each intersection point becomes a vertex, and each road becomes an edge. To reduce the complexity of graph all the intermediate points between two intersections were discarded. After the preprocessing, the graph becomes significantly smaller in size where the total nodes are 129289 and the edges are 281085.

III. Methodology

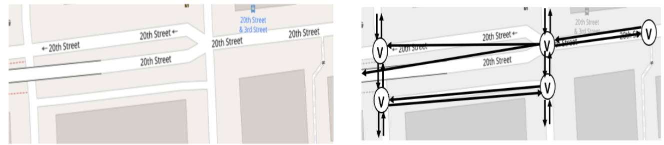
We have utilized weighted directional graph networks to model a large-scale road network. In section 3.1, we discuss our graph model that captures topological and geographical attributes effectively. We also describe how we measure the corresponding edge weights of graph. Based on this graph-based model, we propose a novel deep learning architecture, RoadCaps consolidating GCN and Capsule Networks that estimates the betweenness score of a single/multiple intersections.

A. Modeling Road Networks using Graph Networks [RQ1,2]

Our graph networks reflect the unique set of attributes that comprises topological, physical, and regional characteristics of road networks.

1) Topological Characteristics: We represent the topological attributes of graph components using vertices and edges. The intersections and end points of roads are represented as vertices and the physical roads that connect a pair of vertices are represented as edges.

Topological Characteristics In our model, road networks are represented as a weighted directed graph, $G = (V, E)$,



(a) (b)

Fig. 1. (a): The 20th Street with One-way and Two-way portions (b): Graph representation of the streets depicted in (a)

where the set of vertices, V, represents intersections and end points of the roads, and the set of edges, E represents the physical road between two vertices u and v, where $u, v \in V$. A vertex contains a unique identifier, vertex ID, properties such as geospatial coordinates, connected street count, and road type. An edge is composed of source and destination vertex IDs, and properties including the type of road, weights, length, maximum speed, number of lanes, and road direction. As depicted in fig. 1, the direction of an edge is determined based on the actual traffic flows in the road system. Therefore, if there is a one-way street connecting two intersections (fig. 1-(a)), it will be depicted as a single edge following the direction of the road (fig. 1-(b)). Our graph model considers the geospatial coordinates of the source and destination vertices only.

Weights with Physical Characteristics Our graph representation maintains a vector of features for each vertex. To reflect the tendency of the traffic flow within a road segment, we measure the $traffic\_tendency$ for each edge $e \in E$. We calculate the traffic tendency $e_{traffic\_tendency}$ as follows.

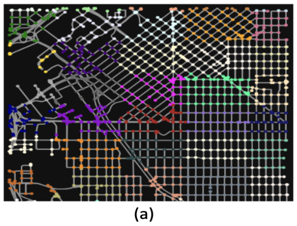
$$e_{traffic\_tendency} = \frac{e_{number\_of\_lanes} \times e_{max\_speed}}{e_{length}} \tag{2}$$

,where $e_{number\_of\_lanes}$ is the number of lanes, e_{max_speed} is the speed limit, and e_{length} is the length of the edge.

A high traffic tendency indicates that the road is designed for high traffic flows. Meanwhile, a low traffic tendency represents that low traffic flow has been expected. Since the shortest path calculation as a part of betweenness estimation gives priority to the path with smaller weight, we have used the inverse of $e_{traffic\_tendency}$ inversely for the edge weight. Besides the traffic tendency, non-numeric properties such as the type of the road (one way or bidirectional), number of incoming roads as indegree, and number of outgoing roads as outdegree from any intersection point are also maintained.

2) Regional Characteristics: Unlike other networks such as social networks, road networks have limited in-degrees and out-degrees of vertices due to physical and topological constraints. This results in well-defined topological patterns across the networks. Therefore, a model cannot factor in the complexity of the regional road system effectively if it targets a smaller radius in the networks. However, inputting entire networks for each estimation would not be a computationally feasible solution for the real-time BC analysis.

To strike a balance between the detecting regional complexity and computational effectiveness, we introduce a complexity



12	16	10	12	4
35	34	40	47	35
13	47	46	49	24
21	48	43	57	39
40	66	50	29	31
23	31	30	27	23
27	29	41	38	22

(b)

Fig. 2. (a): GeoDensity map of road network and (b): intersection count for each Geo-Hash

measure based on the density of streets within a geospatial scope. *GeoDensity* Map is a gridded map with density of intersections within a geohash bounding box. Since the possible number of roads that can share the intersection is not highly variable (only 0.17% of our intersections are shared by 6-8 edges), we define the *GeoDensity* of a vertex as the density of intersections without considering the number of edges.

A geohash is a geospatial encoding system that generates a bounding box identified by a 5-bit character string [18]. The precision of the spatial bounding box is determined by the length of the string identifier. As a greater number of letters are used, the size of the bonding box is reduced. The geohash algorithm provides a hierarchical spatial data structure that preserves the proximity of spatial bounding boxes. We generate a geohash based map with the length of 5 that encompasses approximately 4.9 km^2 in our study areas. All the vertices in the same geohash bounding box share a GeoDensity value. A high GeoDensity value indicates that the given geohash box might be a part of complex road system, therefore more routes are to be expected. On the other hand, a low value signifies a sparse road system. fig. 2 depicts an example of the GeoDensity map with different density of the intersections.

B. Network Architecture [RQ1,2]

Estimating weighted BC values involves multiple factors especially the neighbor road connectivity around the target intersection(s). RoadCaps captures network connectivity and their features by incorporating graph embedding using three layer of Graph Convolutional Network(GCN). The output from GCN is inputted to a Capsule Network layer to capture the hierarchical conceptual structure between the target node(s) and neighboring nodes. fig. 3 depicts the overview of the network architecture. To form the network architecture the first step is to form the graph structure which is described in the following sections.

1) Aggregating Road System Properties with the Graph Structure [RQ1,3]*:* First block of our model leverages graph embedding methods using Graph Convolutional Neural Network (GCN) to incorporate topological connectivity of graphs [19]. Our approach stacks 3 layers to generate graph embeddings. fig. 4 (a) contrasts the model performances with different number of layers, and with 3 layers of GCN, our model shows the better accuracy with less computation complexity.

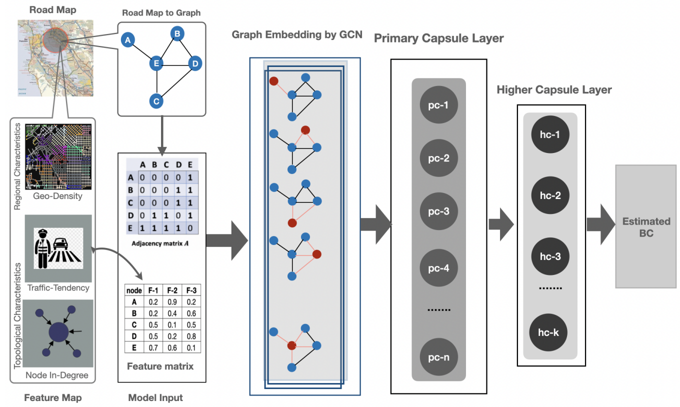


Fig. 3. Proposed Model Architecture (RoadCaps)

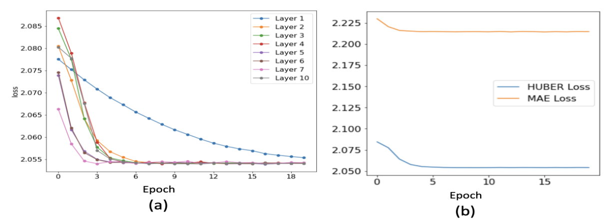


Fig. 4. (a): Model Performance with different number GCN layer (a) and (b): with different loss functions

The first layer of GCN works as the input layer, where the number of neurons equals the number of nodes in the input sample. In this study, we define the maximum number of neurons as 500. The number of neurons indicates the number of neighboring intersections considered within a single input dataset. Since the computing complexity of the BC analysis is closely related to the number of vertices considered for the computation, we have generated small scope of sub-networks to reduce the computing cost. The neighboring intersections are selected based on the distance to the intersection points. Our original dataset from OSM. We have reduced the number of nodes involved by modifying the dataset to include only intersections.

So each input sample has an adjacency matrix of size 500 × 500.Each node has five different features which makes the size of the input features matrix 500 × 5. Then, the very first layer will perform graph convolutions on the feature matrix following the adjacency matrix. Next GCN layers perform the same operation taking the aggregated information in previous layer. Since adding more layers increases the computational complexity, we choose three GCN layers which provides reasonable accuracy for our task.

The dimension of the output of the third layer of GCN maintains the same graph structure as the input sample, $C \times 500$, where C is the number of output channels or filters. Performing efficient feature engineering and selecting effective features is one of the most challenging tasks for road network analysis. To maintain both topological and regional characteristics of the road network along with the road weight, we have also performed feature engineering for our model. The names of the features are traffic tendency computed using eq. 2, GeoDensity, in-degree, and out-degree. Geo density is calculated from the density of connected nodes whose location share the same geohash prefix of five of characters.

2) High Level Features Mapping and BC Estimation with Capsules [RQ3]*:* We use Capsule Networks (CapsNet) to efficiently map these low-level features to high-level features to accurately estimate BC. CapsNets are a neural network architecture that efficiently captures spatial relationships between objects and organizes them in a hierarchical fashion. These objects can represent any spatial pattern (e.g. intersections or a busy highway) and are represented in a vector format known as a capsule. Each value in the capsule represents a different attribute of that object. For example a capsule representing an intersection might have one value representing the direction of an adjacent road and another value representing the number of nearby lanes. Which attributes are stored in a capsule is based purely on what increases the model's accuracy and can be virtually any object attribute. Each layer of a CapsNet stores a set number of capsules, which are then routed to the next Capsule Network layer with its own set of capsules. This dynamic routing process essentially predicts what low-level capsules (e.g. an intersection) from the first CapsNet layer make up the high-level capsules (e.g. a grouping of intersections) in the next CapsNet layer.

CapsNets have been widely applied due to its dynamic routing process being more efficient at capturing information. Also, CapsNets combine the adjacent node values in a non-linear fashion (unlike pooling processes) preserving more spatial information and increasing the model's overall accuracy.

A CapsNet is placed directly after a few GCN layers. This may seem counterintuitive at first, but using only a few convolutional layers to initially encode the data is an efficient way to encode the lowest-level of data and is very much the norm for Capsule Networks.

C. Loss Functions and Hyper Parameters

Our proposed architecture is a single and multi-point regression model. We experimented with the three well-known loss functions for the regression model. These are Mean Squared Error (MSE), Mean Absolute Error (MAE), and Huber loss function. Due to the extensive range and irregularity of target values, MSE does not work well for our model. Both MAE and Huber loss functions fit well into our model. The loss plots of model training for different loss functions are shown in fig. 4 (b) We use the PyTorch L1-loss function that estimates the MAE between each element in the input x and target y. The loss can be described as:

$$\ell(x,y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = |x_n - y_n| \tag{3}$$

$$\ell(x,y) = \begin{cases} \operatorname{mean}(L), & \text{if reduction} = \text{‘mean’;} \\ \operatorname{sum}(L), & \text{if reduction} = \text{’sum’.} \end{cases}$$

For most of the analysis, we use Pytorch HUBER Loss which utilizes a squared term if the absolute element-wise error drops below delta and a delta-scaled L1 term otherwise. This loss unites the benefits of both L1-Loss and MSE-Loss; the delta-scaled L1 region makes the loss less susceptible to outliers than MSELoss, while the L2 part does smoothness over L1Loss near 0. The loss can be described as:

$$\ell(x,y) = L = \{l_1, ..., l_N\}^T \tag{4}$$

with

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Adam is used as the optimizer with an initial learning rate 10^{-4} and a weight decay factor of 10^{-6}. Total GCN layers are 3, and 50 parallel filters are used for GCN layers. RELU is used as the activation function. The number of neurons in each GCN layer and capsules in the primary capsule layer equals the number of total nodes in each sample which is 500 for most experiments. The higher-level capsule layer contains 500 capsules if the target points are 500; otherwise, it equals the number of prediction points. The capsule dimension is set 5 empirically for both capsule layers.

D. Feature Extraction and Selection [RQ2]

The road network can be represented as a large graph where each intersection point of streets is a node, and each road is an edge. Other information is also important to traffic systems like maximum road speed, road direction, street count, and the number of lanes, and geospatial attributes. From the raw information provided, we extract road length, maximum speed, and the number of lanes to construct the feature named traffic tendency. From the topological information, we calculate the degree of incoming and outgoing connections of each node, boundary and non-boundary labeling for each node based on the edge cut in graph samples. We create the geospatial feature, GeoDensity from the geohashes. We consider the first five prefixes of the geohash value to form a group and then count the total number of the intersection points that fall in the same prefix group. The total count of a group is treated as the GeoDensity for all the nodes of that group. After empirical analysis, we select in-degree and out-degree, the traffic tendency, and GeoDensity as the features for each node to train our model. The performance and the time of the model training vary with each feature.

IV. Empirical Benchmarks and Performance Evaluation

A. Experimental Setup

To extract the road network, OSM API v1.1.2 was used. Networkx 2.7.1 was used for preprocessing and graph sampling. Pytorch package 1.10.2 with Python 3.9.10 was used as the machine learning framework.

B. Model Training [RQ3]

The preprocessed road network is converted to a directed graph structure that has 129289 nodes and 281085 edges. The inverted traffic tendency calculated using (2) is used as the edge weight. Since the complexity of exact BC estimation formula is very high ($O(n^3)$), we have used Brandes BC approximation formula to generate the ground truth for this extensive weighted graph [10]. Based on the number path length considered to calculate shortest paths during BC approximation, it takes hours to days. The value range of the ground truth is 10^{12}, as some nodes, mainly in the boundary region, have BC close to zero. We normalized the BC values to 0 and 1000 before feeding the input into the model. After calculating the ground truth, We have generated more than 4000 sub-graphs randomly using the single pivot-based snowball graph sampling technique [20]. , each of which represents a small region of the California road network. From the randomly generated sample graphs, we discarded some very similar samples, and then normalized features were extracted for these sample graphs. We have divided the dataset as 70% for training and 30% for testing after random shuffling. Our models takes few epochs to converge to an optimum point shown in fig. 4 and fig. 7. Each epoch takes around 25 seconds for single target point and 2.26 minutes for 500 target points.

C. Model Performance Analysis

1) Comparisons between models: Model accuracy per model: The range of the ground truth values of the dataset is large enough and the values are not properly distributed from low to high range. The HUBER loss function is found less sensitive to the irregularity of such kind of dataset [21] and so for getting good model performance, we use this loss function for our experiments. We have compared the performance of our model with two other models. One is the base graph convolutional network (GCN) model, and the other model is the GCN with a fully connected layer.

The GCN model is used as the base machine learning model to estimate the improvements we found from our proposed dynamic routing-based learning and capsule-based computation model. To compare the robustness and model strength for multi-point prediction, we have used another model, GCNFCL by adding a linear layer on top of the GCN layer. Comparing the RoadCaps model to GCN model provides a good contrast of the changes in both performance and computational cost when we use capsule layers with GCN. Comparing the performance of GCNFCL and GCN model to RoadCaps gives the effectiveness of the capsules with GCN to the diverse road system. For any size, n, of sample graph, only two distinct types of prediction are possible by the base model GCN, target one, or target n. Keeping the same sample size, n if it is needed to predict centrality for any number of points between one and n, the only way is to add any layer on top of the outer GCN layer that reduces the dimensionality in the output layer. The fully connected layer (FCL) in GCNFCL model and the higher level capsule layer in our proposed RoadCaps model can do this. The fig. 5 (a) shows that our proposed model performs better than the other two models for all the distinct levels of target points. Our model is robust to work with any number of target points. RoadCaps leverages the information-capturing capability of the capsule network. The two layers of capsules following the GCN layers helps the model converge quickly to the optimal point. As depicted in fig. 5 and 7, RoadCaps converges faster with better accuracy than the other two models. The fig. 6 (a) and 6 (b) show that RoadCaps has very stable performance regardless of location and does not very sensitive to boundary effects.

2) Analysis of the importance of features: GCN is well known for its impressive performance in embedding the topological characteristics and aggregation of node features from

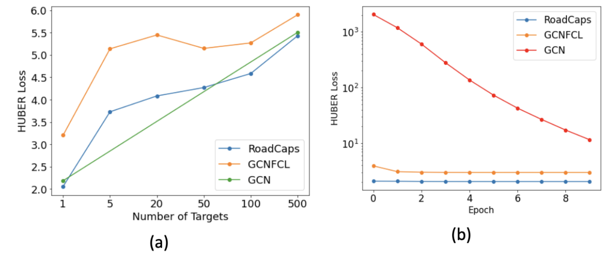


Fig. 5. (a): Model accuracy with different targets and (b): single target

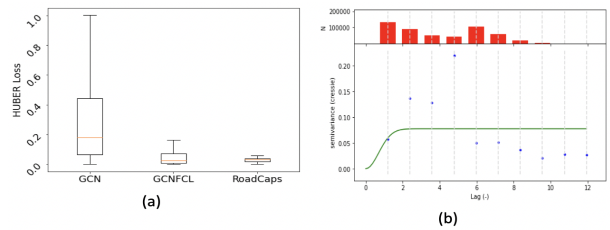


Fig. 6. (a): Model accuracy and (b): variogram of RoadCaps

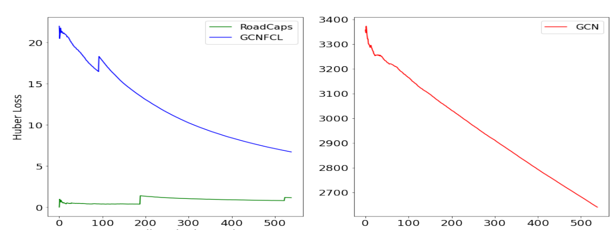


Fig. 7. First epoch of model training

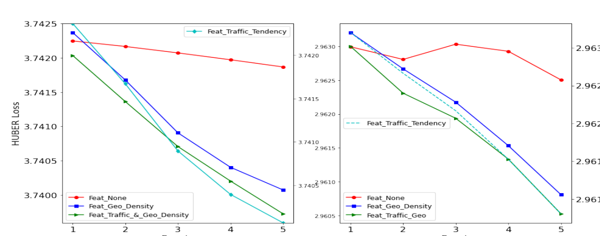


Fig. 8. Model performance for different features

TABLE I
MODEL PERFORMANCE FOR DIFFERENT FEATURES AFTER FIVE EPOCHS

Target Nodes	Without Feature-1 and 2		With Feature-1		With Feature-2		With Feature-1 and 2	
	Train	Test	Train	Test	Train	Test	Train	Test
1	1.145309	2.076185	1.122853	2.054372	1.122862	2.054366	1.124064	2.055077
5	2.962506	3.741867	2.961281	3.740422	2.960803	3.740068	2.960525	3.739724
50	4.256724	4.27026	4.252890	4.270248	4.252888	4.270246	4.252885	4.270243

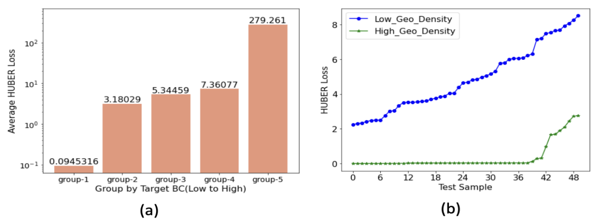


Fig. 9. (a): Model Performance with different BC range and (b) GeoDensity

neighbor nodes of a graph. We also leverage this advantage using GCN block prior to the Capsule layers. Also, we use road network specific features to enhance the model accurcy.fig. 8 and table I show that the GeoDensity and traffic tendency features contribute to model accuracy and faster convergence.

3) Scalability Analysis: The motivation behind using a machine learning model is to estimate the road importance in real-time. At the same time, the model needs to be scalable. Our model performs well for multi-point regression. we observe the performance of the model by changing the number of target points and neighbor nodes. fig. 5 and 10 show that our model is highly robust to the changes in sample size and target points and can maintain better accuracy.

4) Model Performance per BC range: For the road network, it is quite common that the centrality distribution may not be normal or standard as it depends on the road connectivity. The road network dataset of California state also reflects that. Therefore, we have evaluated our model performance for different ranges of BC values. As depicted in fig. 9 (a), we grouped samples for 5 ranges. Most of samples have small BCs (group 1 and 2), and the groups with larger BCs contain higher number of outliers.

D. Geospatial Analysis

1) High density vs Low density area: To observe the model performance with varying road connection density, we have measured the model accuracy for diverse levels of road density. The model performance for 50 samples from each group is shown in fig. 9 (b). The loss values are sorted in ascending order. Our model preserves the reasonable accuracy for both types of road network, but it provides higher accuracy for comparatively complex road network. The BC estimation is closely related to the road connections. We varied the number of neighboring nodes. First, the number of neighbor nodes must be the same or more than the number of target nodes. Secondly, if the sample size is increased, more computation is needed to process all the features and thus the computation complexity of the model is also increased. We did an experiment for a single target node with five different neighbor

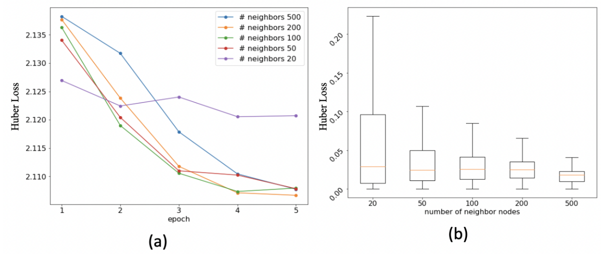


Fig. 10. Model accuracy with different number of neighbor nodes

nodes. fig. 10 shows that sample with at least 50 neighbor nodes provides good accuracy.

V. RELATED WORKS

A. Graph Neural Networks

Recently, graph neural network models (GNNs), which work on graph structured data and apply deep neural network models , have drawn significant interests of the researchers. GNNs capture the dependence of graphs by passing message between the graph nodes [22]. Based on graph embedding and convolutional neural network, the new variants of graph neural networks were mainly proposed to aggregate information collectively from the graph structure. Thus, the input and/or output, consisting of elements and their dependency can be modeled. Several comprehensive reviews on graph neural networks are available. However, these works specifically focus on convolution operators applied on graph structures. The significant part of our proposed work is comprised of graph convolution operation on road networks.

GNN has been used for two types of graph learning problems primarily, graph classification and node classification. Graph classification predicts the class label of graphs. At past, the most dominant graph classification technique were graph kernels [23] but recently, deep learning techniques are getting a big attention [24]. GNNs directly classify graphs depending on the extracted graph representations and that is why GNNs are much more efficient than graph kernel methods [12].

GNN has been applied for the road network analysis such as traffic flow estimation or traffic forecasting [25]. GNN is also used for city-wide parking availability prediction [26]. Our purpose is analysing road network to predict the road importance which then can be used to model any emergency evacuation plan during natural disasters.

One of the biggest challenge of applying GNN in road network analysis is to work on a large graph structure encompassing million to billion of nodes and edges. Researchers propose different ways of using GNN for large scale training. In [27], a method was proposed named DistGNN that optimizes the Deep Graph Library (DGL) to use an efficient shared memory implementation while training on CPU clusters. A minimum vertex-cut graph partitioning algorithm with delayed update has been introduced to reduce the communication latency. Instead of whole-graph based training by Marco et al. [28]

did a case review and provide importance on sample-based training for large scale network.

VI. Conclusion

We present our model, RoadCaps, that estimates BC accurately and rapidly over an extensive road network. RoadCaps addresses model performance challenges emanating from topological complexity and geospatial variability with a custom deep network architecture that incorporates GCN and Deep Capsule Network [RQ1, 2]. GCN captures topological graph structure and features of regional road networks [RQ1], and the Capsule network maps different levels of information and extracts patterns from high-level information effectively [RQ1]. Appropriately extracted topological and geospatial features improve the model accuracy and convergence rate. RoadCaps outperforms our base modeles such as GCN and GCNFCL in terms of accuracy and robustness. Our analysis shows that RoadCaps demonstrates stable performance regardless of the location of target intersection in the sample [RQ2]. RoadCaps trains and makes inferences with compact sized samples that allows computational effectiveness [RQ3].

VII. Acknowledgment

This research was supported by the National Science Foundation (OAC-1931363 and ACI-1553685), the National Institute of Food and Agriculture (COL0-FACT-2019), and a Cochran Family Professorship.

References

[1] H. Tabari, "Climate change impact on flood and extreme precipitation increases with water availability," *Scientific reports*, vol. 10, no. 1, pp. 1–10, 2020.

[2] Gonçalves *et al.*, "Resilience of urban transportation systems. concept, characteristics, and methods.," *Journal of Transport Geography*, vol. 85, p. 102 727, 2020.

[3] Pan *et al.*, "Vulnerability and resilience of transportation systems: A recent literature review," *Physica A: Statistical Mechanics and its Applications*, vol. 581, p. 126 235, 2021.

[4] B. Abdulla, "Characterization of the vulnerability of road networks to fluvial flooding using network percolation approach," *ASCE Computing in Civil Engineering 2019*, 2019.

[5] Kirkley *et al.*, "From the betweenness centrality in street networks to structural invariants in random planar graphs," *Nature communications*, vol. 9, no. 1, pp. 1–12, 2018.

[6] D. J. Aldous, "Routed planar networks," *EJGTA*, vol. 4, no. 1, pp. 42–59, 2016.

[7] M. Barthélemy, "Crossover from scale-free to spatial networks," *EPL*, vol. 63, no. 6, p. 915, 2003.

[8] Crucitti *et al.*, "Centrality measures in spatial networks of urban streets," *Physical Review E*, vol. 73, no. 3, p. 036 125, 2006.

[9] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[10] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of mathematical sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[11] Defferrard *et al.*, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[12] Wu *et al.*, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[13] M. A. Beauchamp, "An improved index of centrality," *Behavioral science*, vol. 10, no. 2, pp. 161–163, 1965.

[14] E. Batschelet, "Circular statistics in biology.," *ACADEMIC PRESS, NY 10003, 1981, 388*, 1981.

[15] Grofman *et al.*, "A game theoretic approach to measuring degree of centrality in social networks," *Social Networks*, vol. 4, no. 3, pp. 213–224, 1982.

[16] Henry *et al.*, "Spatio-temporal correlations of betweenness centrality and traffic metrics," in *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems*, 2019, pp. 1–10.

[17] J. Bennett, *OpenStreetMap*. Packt Publishing Ltd, 2010.

[18] Balkić *et al.*, "Geohash and uuid identifier for multi-agent systems," in *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, Springer, 2012, pp. 290–298.

[19] Berg *et al.*, "Graph convolutional matrix completion," *arXiv:1706.02263*, 2017.

[20] L. A. Goodman, "Snowball Sampling," *The Annals of Mathematical Statistics*, vol. 32, no. 1, pp. 148–170, 1961.

[21] Esmaeili *et al.*, "A novel approach to quantized matrix completion using huber loss measure," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 337–341, 2019.

[22] Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[23] Navarin *et al.*, "Approximated neighbours minhash graph node kernel.," in *ESANN*, 2017, pp. 281–286.

[24] Narayanan *et al.*, "Graph2vec: Learning distributed representations of graphs," *arXiv:1707.05005*, 2017.

[25] Yu *et al.*, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv:1709.04875*, 2017.

[26] Zhang *et al.*, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1186–1193.

[27] Md *et al.*, "Distgnn: Scalable distributed training for large-scale graph neural networks," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14.

[28] Serafini *et al.*, "Scalable graph neural network training: The case for sampling," *ACM SIGOPS Operating Systems Review*, vol. 55, no. 1, pp. 68–76, 2021.