IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 9, 2019, accepted May 5, 2019, date of publication May 9, 2019, date of current version May 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915974

An Efficient Approximation of Betweenness Centrality for Uncertain Graphs

CHENXU WANG, (Member, IEEE), AND ZIYUAN LIN
School of Software Engineering, MoE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Chenxu Wang (cxwang@mail.xjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation under Grant 61602370, Grant 61672026, Grant 61772411, and Grant U1736205, in part by the Postdoctoral Foundation under Grant 201659M2806 and Grant 2018T111066, in part by the Fundamental Research Funds for the Central Universities under Grant 1191320006, in part by the Shaanxi Postdoctoral Foundation through SZSTI under Project JCYJ20170816100819428, in part by the CCF-Tencent Open Fund WeBank Special Funding under Grant CCF-Webank RAGR20180101, and in part by the CCF-NSFOCUS KunPeng Research Fund under Grant CCF-NSFOCUS 2018006.

ABSTRACT Betweenness centrality measures the centrality of nodes and edges in a graph based on the concept of shortest paths. However, such a definition is unsuitable for uncertain graphs due to the uncertainty of links. In the possible-world semantics, the Monte Carlo method is proposed to estimate the betweenness centrality of uncertain graphs. However, this method is computationally intensive. To address this challenging issue, in this paper, we propose the concept of possible shortest paths and develop a metric to approximate the betweenness centrality for uncertain graphs. We demonstrate that the new metric of betweenness centrality generalizes the deterministic one. Unfortunately, it is NP-hard to enumerate all possible shortest paths between two nodes exhaustively. To tackle this difficulty, we design a heuristic algorithm to explore the majority of possible shortest paths efficiently. Our method avoids the sampling process in the Monte Carlo method, and thus significantly improves the computational efficiency. We conduct extensive experiments to evaluate the effectiveness and efficiency of our method. The experimental results show that our approach can approximate the centrality of uncertain graphs accurately with high efficiency. Finally, we apply our method to the Internet network to evaluate the importance of autonomous systems.

INDEX TERMS Uncertain graphs, betweenness centrality, possible shortest paths, connectivity.

I. INTRODUCTION

Many network-based systems such as the Internet, social networks, and biological networks are inherently dynamic. Consequently, link uncertainty arises as the network evolves, e.g., links observed at an earlier time may no longer be present or active at the time of analysis [1]. In addition, noisy measurements, inference models, and privacy preserving perturbation processes also produce uncertain link data [2]. The probabilistic graph model has been proposed to capture the uncertainty of links by associating each link with a presence probability [3].

With the introduction of link uncertainty,[1] conventional centrality measurement methods such as node degrees [4], average length of the shortest paths [5], betweenness centrality [6], clustering coefficients [7], and pairwise connectivity [8] may not be capable for the assessment of probabilistic graphs. Dinh and Thai investigated the vulnerability assessment by measuring the expected pairwise connectivity of probabilistic graphs [3]. By treating a probabilistic graph as a generative model of a set of deterministic graphs which are possible realizations of the probabilistic graph, they formulated the problem as a stochastic optimization problem. They proposed a Fully Polynomial Time Randomized Approximation Scheme to estimate the expected pairwise connectivity with any desired accuracy. However, the method is still too computationally intensive to measure large-scale uncertain graphs.

Pfeiffer and Neville studied the measures of path lengths, betweenness centrality, and clustering coefficients for probabilistic graphs [1]. They developed a probability-based algorithm to calculate these measures. However, they only consider the most probable paths. It underestimates the metrics because some neglected possible paths may also contribute to the measures. For instance, Fig. 1 presents a toy example

The associate editor coordinating the review of this manuscript and approving it for publication was Saad Bin Qaisar.

[1] In this paper, we use the terms node and vertex, edge and link, graph and network interchangeably.

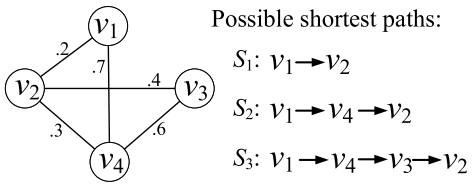

FIGURE 1. An example of a probabilistic graph.

of a probabilistic graph with presence probabilities illustrated on the edges. There are three possible paths between v_1 and v_2, namely S_1, S_2, and S_3. The conditions for the paths S_1, S_2, and S_3 to be the shortest one are dependent. For instance, the condition for S_2 to be the shortest path relies on the absence of S_1, and the condition for S_3 to be the shortest path depends on the absence of both S_1 and S_2. In addition, v_1 and v_2 also have a probability to be disconnected. Ignoring the dependency and disconnection possibility may result in unpredictable biases.

In this paper, we propose a novel approach to efficiently approximating the betweenness centrality of probabilistic graphs. Motivated by the possible-world semantics in the computation of probabilistic graphs [2], [9], we define the concept of possible shortest paths between two nodes. Each possible shortest path has a probability of being the shortest one between the two nodes in one of the realizations, e.g., S_1, S_2, and S_3 as illustrated in Fig. 1. Additionally, the probabilities of these paths are inter-dependent (i.e., a path to be the shortest one depends on the absence of paths which are shorter than it).

Based on the definition of possible shortest paths, we formally define the betweenness centrality for probabilistic graphs. Similar to the conventional definition of betweenness centrality for deterministic graphs which measures the sum of the fraction of all-pairwise shortest paths that pass through a node or an edge, our definition measures the sum of the fraction of all-pairwise possible shortest paths that pass through a node or an edge. Besides, the contribution of a possible shortest path to the computation of betweenness centrality is relevant with its probability to be the shortest path in a realization of the probabilistic graph. We demonstrate that the new definition of betweenness centrality is a general case of the deterministic one.

However, it is time-consuming to explore all possible shortest paths between two nodes exhaustively. To tackle this issue, we develop a heuristic algorithm to improve efficiency. Instead of exploring all possible shortest paths, the algorithm uncovers those paths with high probabilities. It significantly improves the computational efficiency while maintains high accuracy. Then we develop an efficient algorithm to approximate the betweenness centrality of a probabilistic graph.

Finally, we conduct extensive experiments to validate the effectiveness and efficiency of our method. Compared with the state-of-the-art approaches, our method is both effective and efficient in estimating the betweenness centrality of homogeneous and heterogeneous probabilistic graphs.

In summary, we make the following contributions:

1) To the best of our knowledge, we are the first to propose the concept of possible shortest paths and formally define the possible shortest path based betweenness centrality for probabilistic graphs. The definition generalizes the deterministic betweenness centrality and measures the centrality of nodes and edges for probabilistic graphs.
2) We design a heuristic algorithm to efficiently explore possible shortest paths between two nodes in a probabilistic graph. We also develop an efficient algorithm to approximate the betweenness centrality of a probabilistic graph. Compared with the Monte Carlo method, our algorithm avoids the computationally intensive sampling process and thus is more efficient.
3) We evaluate the effectiveness and efficiency of the proposed method. The experimental results show that our method outperforms state-of-the-art methods.

The rest of this paper is organized as follows. In Section II, we introduce the probabilistic graph model and its computation. In Section III, we present the definitions of possible shortest paths and probabilistic betweenness centrality for probabilistic graphs. In Section IV, we develop algorithms to compute the betweenness centrality efficiently. In Section V, we conduct experiments to validate the effectiveness and efficiency of our method. After reviewing the relevant work in Section VII, we conclude this paper in Section VIII.

II. PROBABILISTIC GRAPH MODEL

Denote a probabilistic graph by $\mathcal{G}(V, E, P)$, where V represents the set of nodes; $E \subset V \times V$ corresponds to the set of edges; and P contains the probabilities associated with the edge set E. Let $\Pr[(v_i, v_j)] \in [0, 1]$ be the probability of the edge between v_i and v_j; $\Pr[(v_i, v_j)] = 0$ if there is no edge and $\Pr[(v_i, v_j)] = 1$ if there is a definite edge between v_i and v_j. In the rest of this paper, we also use p_{ij} as an alternative of $\Pr[(v_i, v_j)]$ for convenience if it does not confuse. We assume that edges are independent of one another.

In the possible-world semantics, a probabilistic graph $\mathcal{G}$ is viewed as a generative model for discrete deterministic graphs. A deterministic graph $G_s(V, E_s)$ can be generated from $\mathcal{G}$ by independently sampling each edge $e_{ij} \in E$ with the associated probability p_{ij}. We refer to $G_s \sqsubset \mathcal{G}$ as a realization or a sample of $\mathcal{G}$. Since edge independence is assumed, the probability of G_s can be calculated by:

$$\Pr(G_s) = \prod_{e_{ij} \in E_s} p_{ij} \prod_{e_{ij} \in E \setminus E_s} (1 - p_{ij}). \tag{1}$$

Each G_s is a possible world of $\mathcal{G}$ and there are totally $2^{|E|}$ possible worlds. Given a measure ϕ (e.g., betweenness centrality, clustering coefficients), let $\phi(G_s)$ denote the value of the measure obtained from a sampled graph G_s. Then, we can calculate the *expected value* of the measure for a probabilistic graph $\mathcal{G}$ by enumerating all the possible worlds

G_s and applying the formulation:

$$\hat{\phi} = \mathbb{E}[\phi(\mathcal{G})] = \sum_{G_s \sqsubseteq \mathcal{G}} \phi(G_s) \Pr(G_s) \tag{2}$$

However, computation of the expected value needs enumerate all possible worlds which grows exponentially with the number of uncertain edges. It is intractable for graphs with even some dozens of uncertain edges.

A common approach to overcoming the intractability of computing the expected measure is the Monte Carlo (MC) method which consists of the following steps:

1) Randomly sample r discrete graphs from the probabilistic graph according to P;
2) Compute the values of the measure for all sampled graphs;
3) Approximate the expected value of the metric by calculating the mean of the obtained values.

The Chernoff bound theorem [1] guarantees that the calculated mean is an unbiased estimate of the concerned measure:

Theorem 1 (Chernoff Bound): Let $X_1, X_2, \ldots, X_r$ be independent and identically distributed random variables bounded by the interval $[0, 1]$. *We define the empirical mean of these variables by* $\bar{X} = \frac{1}{r}\sum_{i=1}^{r} X_i$. *The expectation of these variables is* $\mu = E[X_i]$. *If* $r \geq \frac{3}{\epsilon^2 \mu} \ln(\frac{2}{\delta})$, *then we have* $\Pr\left(|\bar{X} - \mu| \geq \varepsilon\mu\right) \leq \delta$. *We say that the r samples provide with an (ε, δ)-approximation of μ.*

Theorem 1 guarantees that we can obtain an unbiased estimate of a measure without enumerating all possible worlds of a probabilistic graph. However, the sampling process of a probabilistic graph is still computationally intensive. For instance, let $\mu \approx 0.1$, we have to do at least 15,895 samples to provide with an $(0.1, 0.01)$ approximation of μ. To address this issue, in this paper, we develop a novel approach to estimating the betweenness centrality of a probabilistic graph efficiently.

III. DEFINITIONS

In this section, we first present the concept of possible shortest paths and then formally define the betweenness centrality for probabilistic graphs.

A. POSSIBLE SHORTEST PATHS

For deterministic graphs, the shortest paths between two nodes are definite. However, for probabilistic graphs, since there are many possible realizations of a probabilistic graph, the shortest paths between two nodes in different realizations are not necessarily identical. There are many possible shortest paths between two nodes in a probabilistic graph. In this paper, we define the possible shortest paths between two nodes in a probabilistic graph as follows:

Definition 1 (Possible Shortest Paths): Given a probabilistic graph $\mathcal{G}(V, E, P)$ and a node pair (v_i, v_j), the possible shortest paths between v_i and v_j is defined as a set $\mathcal{P}(v_i, v_j) = \{S_1, S_2, \ldots, S_{H_{ij}}\}$ constituted by all shortest paths existing in any possible realizations of $\mathcal{G}$ between v_i and v_j.

In the rest of this paper, we also write $\mathcal{P}(v_i, v_j)$ as $\mathcal{P}_{ij}$ for simplicity if no ambiguity arises. $H_{ij} = |\mathcal{P}_{ij}|$ is the number of all possible shortest paths between v_i and v_j. $S_k = [e_1, e_2, \cdots, e_{|S_k|}]$ is one of the possible shortest paths represented by a sequence of connected edges; and $|S_k|$ represents the length of the path S_k.

As mentioned in Section II, each possible realization of a probabilistic graph has a presence probability. Then each shortest path in a possible realization also has an associated probability. Since edges are independent of one another, motivated by the implication of possible worlds, we define the absolute probability of a path S_k as the product of edge probabilities in S_k:

$$\Pr(S_k) = \prod_{e \in S_k} \Pr(e). \tag{3}$$

The absolute probability of a possible path does not depend on the existence of other paths. Take the probabilistic graph in Fig. 1 as an example, there are three possible shortest paths between v_1 and v_2, namely:

$$\begin{aligned} S_1 &= [(v_1, v_2)], \\ S_2 &= [(v_1, v_4), (v_4, v_2)], \\ S_3 &= [(v_1, v_4), (v_4, v_3), (v_3, v_2)], \end{aligned}$$

respectively. The absolute probabilities for the three paths are:

$$\begin{aligned} \Pr(S_1) &= 0.2, \\ \Pr(S_2) &= 0.7 \times 0.3 = 0.21, \\ \Pr(S_3) &= 0.7 \times 0.6 \times 0.4 = 0.168, \end{aligned}$$

respectively. However, the probability for a possible path to be the shortest one in a realization depends on the non-existence of other shorter possible paths. Formally, the relative probability of S_k to be the shortest path in a realization of $\mathcal{G}$ is defined as:

$$\widehat{\Pr}\left[S_k | S_k \in \mathcal{S}_{ij}\right] \approx \prod_{S_l \in \mathcal{P}_{ij}, |S_l| < |S_k|} [1 - \Pr(S_l)] \Pr(S_k). \tag{4}$$

That is, the relative probability of a path to be the shortest one is dependent on the existence of other paths shorter than it. It is worth noting that a path may share links with the shorter one, e.g., the path S_3 shares a common link (v_1, v_4) with S_2. Therefore, the right part in (4) is only an approximation of the relative probability. However, we prove that the approximation has a bounded error with 0.083 (See the Appendix). The relative probability of the shortest path in $\mathcal{P}_{ij}$ equals to its absolute probability. The relative probabilities of other paths in $\mathcal{P}_{ij}$ can be calculated recursively. For instance, the relative probabilities of the three paths in Fig. 1 can be calculated as:

$$\begin{aligned} \widehat{\Pr}(S_1) &= \Pr(S_1) = 0.2, \\ \widehat{\Pr}(S_2) &\approx (1 - \Pr(S_1)) \Pr(S_2) = 0.168, \\ \widehat{\Pr}(S_3) &\approx (1 - \Pr(S_1))(1 - \Pr(S_2)) \Pr(S_3) = 0.106 \end{aligned}$$

It is worth noting that there is also a probability that two nodes are disconnected. We employ Λ_{ij} to represent a non-exist path between v_i and v_j, and $|\Lambda_{ij}| = \infty$. Then the probability that v_i and v_j are disconnected can be calculated by:

$$\widehat{\Pr}(\Lambda_{ij}) = \prod_{S_k \in \mathcal{P}_{ij}} (1 - \Pr(S_k)) \tag{5}$$

We define the connectivity of two nodes as follows:

Definition 2 (Probabilistic Connectivity): Given a probabilistic graph $\mathcal{G}(V, E, P)$ and a node pair (v_i, v_j), the probabilistic connectivity φ_{ij} of v_i and v_j is the probability that there is at least one path between v_i and v_j in an arbitrarily selected realization of $\mathcal{G}$.

Accordingly, the connectivity of v_i and v_j can be calculated by:

$$\varphi_{ij} = 1 - \widehat{\Pr}(\Lambda_{ij}) = 1 - \prod_{S_k \in \mathcal{P}_{ij}} (1 - \Pr(S_k)) \tag{6}$$

For example, the connectivity of nodes v_1 and v_2 in Fig. 1 is

$$\varphi_{12} = 1 - (1 - \Pr(S_1))(1 - \Pr(S_2))(1 - \Pr(S_3)) = 0.474,$$

indicating that v_1 and v_2 have a probability of 0.474 to be connected in an arbitrarily selected realization of the probabilistic graph. Obviously, if the absolute probability of a possible shortest path is 1.0, then $\widehat{\Pr}(\Lambda_{ij}) = 0$ and $\varphi_{ij} = 1$, indicating a definite connection between v_i and v_j.

B. BETWEENNESS CENTRALITY

The betweenness centrality of a node v for a deterministic graph $G(V, E)$ is defined as:

$$c_B(v) = \frac{2}{(|V| - 1)(|V| - 2)} \sum_{s,t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)}, \tag{7}$$

where $\sigma(s, t|v) = \sum_{S_k \in \mathcal{P}_{st}} \mathbf{1}_{S_k}(v)$ is the number of the shortest paths between s and t passing through the node v; $\mathcal{P}_{st}$ is the set of the shortest paths between s and t; $\sigma(s, t) = |\mathcal{P}_{st}|$ is the total number of the shortest paths between s and t; $|V|$ is the number of nodes in $G(V, E)$.

Without loss of the generality, we define the betweenness centrality for probabilistic graphs based on the definition of possible shortest paths.

Definition 3 (Betweenness Centrality of a node for Probabilistic Graphs): Given a probabilistic graph $\mathcal{G}(V, E, P)$ and a node v, the probabilistic betweenness centrality of v is defined as:

$$c_B(v) = \frac{2}{(|V| - 1)(|V| - 2)} \sum_{s,t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)} \varphi_{st}, \tag{8}$$

where $\sigma(s, t|v) = \sum_{S_k \in \mathcal{P}_{st}} \mathbf{1}_{S_k}(v) \widehat{\Pr}(S_k)$ is the sum of relative probabilities of possible shortest paths between s and t that pass through the node v; $\mathbf{1}_{S_k}(v)$ is the indicator function which is 1 when S_k passes through v, and 0 otherwise; $\sigma(s, t) = \sum_{S_k \in \mathcal{P}_{st}} \widehat{\Pr}(S_k)$ is the sum of relative probabilities of all possible shortest paths between s and t; φ_{st} is the connectivity of the nodes s and t; and $|V|$ is the number of nodes in $\mathcal{G}(V, E, P)$.

According to the definition, the betweenness centrality of a node in probabilistic graphs measures the fraction of relative probabilities of possible shortest paths passing through the node. Similarly, we can define the probabilistic betweenness centrality of an edge as:

$$c_B(e) = \frac{2}{(|V|)(|V| - 1)} \sum_{s,t \in V} \frac{\sigma(s, t|e)}{\sigma(s, t)} \varphi_{st}, \tag{9}$$

where $\sigma(s, t|e) = \sum_{S_k \in \mathcal{P}_{st}} \mathbf{1}_{S_k}(e) \widehat{\Pr}(S_k)$ is the sum of relative probabilities of possible shortest paths between s and t that pass through the edge e; $\mathbf{1}_{S_k}(e)$ is the indicator function which is 1 when S_k passes through the edge e, and 0 otherwise; $\sigma(s, t) = \sum_{S_k \in \mathcal{P}_{st}} \widehat{\Pr}(S_k)$ is the sum of relative probabilities of all possible shortest paths between s and t; φ_{st} is the connectivity of the nodes s and t; and $|V|$ is the number of nodes in $\mathcal{G}(V, E, P)$.

It is worth noting that the probabilistic definition of betweenness centrality is consistent with deterministic one. When the presence probabilities of all edges in a probabilistic graph equal to 1.0, indicating a deterministic graph, Equation 7 degrades into Equation 8 since $\widehat{\Pr}(S_k) = 1.0$ and $\varphi = 1.0$.

IV. ALGORITHMS

In this section, we first develop a heuristic algorithm to explore possible shortest paths between two nodes efficiently. Then we present the algorithm to calculate the betweenness centrality for probabilistic graphs.

A. POSSIBLE SHORTEST PATH EXPLORATION ALGORITHM

Determining the connection probability of two nodes in a probabilistic graph has been shown as a #P-complete problem [10]. As a consequence, finding all possible shortest paths between two nodes in a probabilistic graph is also a #P-hard problem. To address this issue, we develop a heuristic algorithm to explore possible shortest paths between two nodes in a probabilistic graph efficiently. The pseudo-code is presented in Algorithm 1, which includes the following four major steps:

1) We view the probabilistic graph as a deterministic graph by ignoring the probabilities of edges (Line 1) and apply any shortest path algorithms (e.g., the Dijkstra algorithm or the breadth-first search algorithm) on the deterministic graph to find all the shortest paths between two nodes (Line 2).
2) For each of the shortest paths returned, we add the path into $\mathcal{P}_{st}$. Then we find the edge with the smallest probability within the path and remove it from the deterministic graph. (Lines from 4 to 7).
3) We find the shortest paths between the two nodes in the modified deterministic graph (Line 8) and calculate the connectivity of s and t based on $\mathcal{P}_{st}$ (Line 9).

Algorithm 1 Possible shortest path exploration algorithm

Input: A probabilistic graph $\mathcal{G}(V, E, P)$; the source node s and the destination node t; a predefined threshold θ.
Output: $\mathcal{P}_{st}$: A set of possible shortest paths
Initialization: $\mathcal{P}_{st} = \emptyset$; $\varphi_{st} = 0$
1 $G \leftarrow$ ignoring the probabilities in $\mathcal{G}$;
2 $\mathcal{P}' \leftarrow$ all the shortest paths between s and t in G;
3 **while** $\varphi_{st} < \theta$ & $\mathcal{P}' \neq \emptyset$ **do**
4 **foreach** S *in* $\mathcal{P}'$ **do**
5 Add S into $\mathcal{P}_{st}$;
6 $e_{\min} \leftarrow$ edge with the smallest probability in S;
7 Remove $e_{\min}$ from G;
8 $\mathcal{P}' \leftarrow$ all the shortest paths between s and t in G;
9 $\varphi_{st} \leftarrow$ connectivity of s and t based on $\mathcal{P}_{st}$;

Algorithm 2 Betweenness centrality algorithm

Input: A probabilistic graph $\mathcal{G}(V, E, P)$
Output: $\mathcal{M}_{node}$: A node map; $\mathcal{M}_{edge}$: An edge map
Initialization: $\mathcal{M}_{node} = \{\}$; $\mathcal{M}_{edge} = \{\}$
1 **foreach** *node pair* (s, t) **do**
2 $\mathcal{P}_{st} \leftarrow$ possible shortest paths between s and t;
3 $\varphi_{st} \leftarrow$ connectivity of s and t based on $\mathcal{P}_{st}$;
4 **foreach** $S \in \mathcal{P}_{st}$ **do**
5 **foreach** $v \in \text{Intr}(S)$ **do**
6 update $\mathcal{M}_{node}(v)$ with $\frac{\widehat{\Pr}(S)}{\sigma(s,t)}\varphi_{st}$
7 **foreach** $e \in S$ **do**
8 update $\mathcal{M}_{edge}(e)$ with $\frac{\widehat{\Pr}(S)}{\sigma(s,t)}\varphi_{st}$
9 Normalize the values of $\mathcal{M}_{node}$ with $(n-1)(n-2)/2$;
10 Normalize the values of $\mathcal{M}_{edge}$ with $n(n-1)/2$;

4) Repeat Step 2 to 3 until one of the following conditions is satisfied (Line 3): i) There are no paths between s and t due to edge removals; ii) The connectivity between s and t based on the current set of obtained possible shortest paths is greater than a predefined threshold θ, e.g., $\theta = 0.99$.

The time complexity of Algorithm 1 depends on the shortest path algorithm selected to explore the shortest paths. For unweighted undirected graphs, the time complexity of the breadth-first search algorithm is $O(|V| + |E|)$. If an average of $\bar{d}$ loops are required for the algorithm, the time complexity of the heuristic algorithm is $O[\bar{d}(|V| + |E|)]$. Intuitively, the value of $\bar{d}$ is relevant to the distribution of edge probabilities. For deterministic graphs, we have $\bar{d} = 1$. Our algorithm is guaranteed to converge because we set a connectivity threshold of θ. We can also further to improve the efficiency of our algorithm by setting a threshold of the maximum number of possible paths between two nodes. However, it sacrifices some calculation accuracy.

B. BETWEENNESS CENTRALITY ALGORITHM

Based on the possible shortest path explored, we then develop an algorithm to calculate the betweenness centrality of a probabilistic graph. The pseudo-code is presented in Algorithm 2, which contains the following major steps:

1) For each node pair, we calculate the set of possible shortest paths based on Algorithm 1. We then calculate the connectivity of the two nodes (Lines from 1 to 3).
2) For each possible shortest path, we traverse every node and edge in the path and update the betweenness centrality of the corresponding nodes and edges (Lines from 4 to 8).
3) Repeat Step 1 to 2 until all node pairs are traversed.
4) Normalize the betweenness centrality of nodes and edges (Lines from 9 to 10).

More precisely, we first calculate the possible shortest paths between all node pairs and cumulatively update the betweenness centrality of the internal nodes and edges. Finally, we normalize the calculated values. The time complexity of Algorithm 2 is $O[\bar{d}(|V|+|E|)|V|^2/2]$. Algorithm 2 calculates the betweenness centrality of an uncertain graph. In each loop, it calculates the set of all possible shortest paths between a node pair by invoking Algorithm 1 and cumulatively updates the betweenness centrality. The algorithm terminates when all node pairs are traversed. However, we believe that the efficiency of our algorithm could be improved by randomly sampling a fixed number of node pairs. Riondato et al. [11] proposed an efficient randomized algorithm for betweenness estimation in deterministic graphs. The algorithm approximates betweenness centrality with the desired accuracy and confidence by calculating the shortest paths of a fixed number of randomly sampled node pairs. The number of samples needed does not depend on the number of nodes in the graph. However, it is still an open problem whether the sampling method is suitable for uncertain graphs in the sense of possible shortest paths. More theoretical analysis is needed before we apply the scheme to uncertain scenarios. We will explore this research direction in our future work.

It is worth noting that Algorithm 1 cannot guarantee to obtain all the possible shortest paths for any node pairs. For example, in Fig. 2, the algorithm will not find the path $S_2 = [(v_1, v_2), (v_2, v_4), (v_4, v_3)]$ between v_1 and v_3 if $p_1 \leq p_2$ since the edge (v_1, v_2) is removed after the algorithm identifies the primary shortest path $S_1 = [(v_1, v_2), (v_2, v_3)]$. However, we argue that such a failure does not affect the results dramatically since the error is up-bounded by a small value. Besides, the exceptional case presented in Fig. 2 leads to the maximum error amongst all exceptional cases. We present the rigorous theoretical proof of the claim in the Appendix.

V. EXPERIMENTS

In this section, we evaluate the effectiveness and efficiency of our method in assessing the centrality of probabilistic graphs.

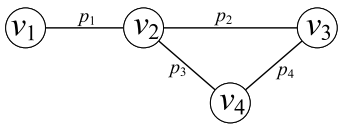

FIGURE 2. An exceptional case where algorithm 1 fails finding a possible shortest path from v_1 to v_3 if $p_1 < p_2 \leq 1$.

TABLE 1. Default parameter settings.

Context	Parameters	Default
ER model	n	500
	p	0.04
BA model	n	500
	m	5
Probabilistic graph	$p_{\min}$	0.0
Algorithm 1	θ	0.8

A. EXPERIMENTAL SETTINGS

We conduct the experiments based on artificial networks generated by two classical network models, namely the Erdős-Rényi model (ER model) and the Barabási-Albert preferential attachment model (BA model). We select these two models because they generate homogeneous (random) and heterogeneous (scale-free) networks, respectively [12]. Given the number of nodes n, the ER model generates a graph by choosing each of the possible edges with a fixed probability p. Given the number of nodes n and the number of edges m to attach from a new node to existing nodes, the BA model generates a graph whose node degree follows a power-law distribution. Given a network generated, we assign each edge with a probability uniformly drawn from the range $[p_{\min}, 1]$, where $0 \leq p_{\min} \leq 1$ is a parameter to evaluate the impacts of the lower bounds of edge probabilities. The threshold θ is used to indicate when the algorithm terminates the exploration process of possible shortest paths between two nodes. Large θ leads to more substantial coverage of possible shortest paths in the output of Algorithm 1. According to our experiments, when $\theta \approx 0.8$, our algorithm achieves the best performance. Hence, we set $\theta = 0.8$ the default value. In the following experiments, we use the default parameter settings listed in Table 1 unless it is pointed out. To guarantee the statistical significance, we repeat the experiments 50 times and calculate the average of the results unless it is pointed out.

B. EFFECTIVENESS VALIDATION

We validate the effectiveness of our method by evaluating the impacts of the threshold θ, the edge probabilities of probabilistic graphs, and the network properties such as the number of nodes and edges (or densities).

1) EVALUATION METRICS

To evaluate the effectiveness of our method, we employ the Monte Carlo (MC) method as a baseline. Given a probabilistic network with parameters n and m, we sample $r = \frac{2}{\varepsilon^2} \ln \frac{2}{\delta}$ realizations of the network. Then we calculate the betweenness centrality of the deterministic realization using Ulrik Brandes' algorithm [13]. In the experiments, we set $\varepsilon = 10^{-2}$ and $\delta = 0.05$, respectively. To evaluate the divergence of an estimation method from the MC approach, we employ two metrics including the mean absolute error (MAE) and Spearman Correlation Coefficient (SCC). Explicitly, given an uncertain graph $\mathcal{G}$, we calculate the betweenness centrality for each node with the MC method and the evaluating method, respectively. Suppose the values calculated by the two methods for node v are B_v^{MC} and B_v^*, respectively, the MAE is defined as

$$MAE = \frac{1}{n} \sum_{v \in \mathcal{G}} |B_v^{MC} - B_v^*|. \tag{10}$$

where n is the number of nodes. This metric evaluates the absolute errors of a method compared with the MC method.

SCC is also a useful metric to evaluate the effectiveness. The primary goal of calculating the betweenness centrality is to evaluate the importance of nodes. Thus, the orders of nodes are somewhat more essential than the absolute values of node betweenness centrality. The SCC metric evaluates the order fluctuations of nodes with the MC method as a baseline. To calculate the SCC, we sort the nodes according to the obtained betweenness centrality. Let X_v^{MC} and X_v^* be the ranks of node v according to the MC method and the evaluating method, then the SCC can be calculated as

$$SCC = 1 - \frac{6 \sum_{v \in \mathcal{G}} (X_v^* - X_v^{MC})^2}{n(n^2 - 1)} \tag{11}$$

where n is the number of nodes. The SCC metric evaluates how well an evaluating method to rank the nodes with the MC method as the baseline.

2) IMPACTS OF THE THRESHOLD θ

In this experiment, we evaluate the impacts of the parameter θ on the effectiveness of our method. The parameter θ is the connectivity threshold to indicate when the algorithm terminates with acceptable accuracy. A more significant value of θ ensures a more substantial coverage of all possible shortest paths between a node pair. Fig. 3 presents the violin plot of calculated absolute errors. The top bars in the figures represent the maximum errors, and the inner bars show the mean of the errors. Besides, the outer shape represents all possible values, and the thickness indicates how common the value is. We find that our method has a very small MAE for both network models. Moreover, most errors concentrate in the range from 0 to 0.005 even θ is small (e.g., 0.09). It demonstrates the effectiveness of our method. Additionally, the MAE and maximum error decrease with the increase of θ when $\theta < 0.79$. However, when $\theta > 0.79$, the MAE and maximum error increase slightly versus θ. It strongly indicates that a very large θ may over-estimate the betweenness centrality. Comparisons between the two network model show that the maximum errors for the BA model are much larger than that for the ER model. It is

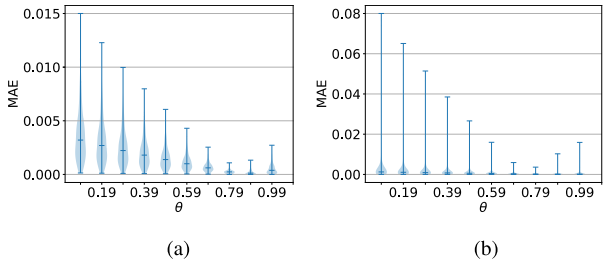

FIGURE 3. Violin plot of absolute errors versus θ. (a) ER model. (b) BA model.

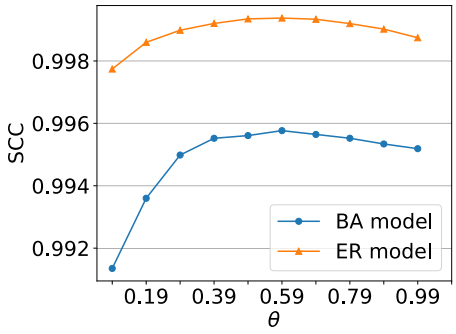

FIGURE 4. SCC versus θ.

because the BA model has more exceptional structures shown in Fig. 2.

Fig. 4 presents the Spearman Correlation Coefficients. The results show that the SCC is always greater than 0.99 for both network models, indicating high effectiveness of our method in ordering the nodes according to their centrality in the possible-world semantics. However, analogous to MAE, SCC is not always monotonically increasing versus the increase of θ. When $\theta > 0.79$, our method over-estimates the betweenness centrality, resulting in a lower SCC. Comparisons between the results of the two network models show that our method performs better for the ER model than the BA model because the BA model is easier to form the exceptional structures.

3) IMPACTS OF EDGE PROBABILITIES

In this experiment, we evaluate the impacts of edge probabilities on the effectiveness of our possible-shortest-path based method (PSP). To further demonstrate the advance of our method, we compare our method with the most probable path method (MPP) [1]. The MPP method computes the most likely paths in much the same way that shortest paths are computed on weighted discrete graphs (edge weights are calculated by $-\log(p_{ij})$), by applying the Dijkstra's shortest path algorithm. Instead of expanding on the shortest path, the method expands the most probable path. In the experiment, we adjust the lower bound $p_{\min}$ of the edge probability in the probabilistic graphs. Then we apply the PSP and MPP methods on the graph and calculate the SCC for both methods.

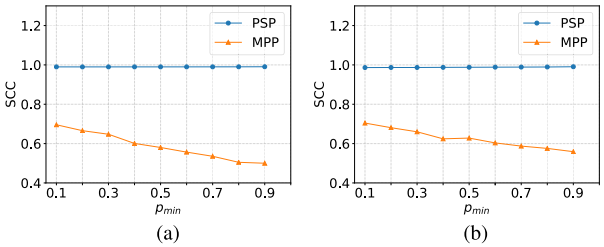

FIGURE 5. SCC versus $p_{\min}$. (a) ER model. (b) BA model.

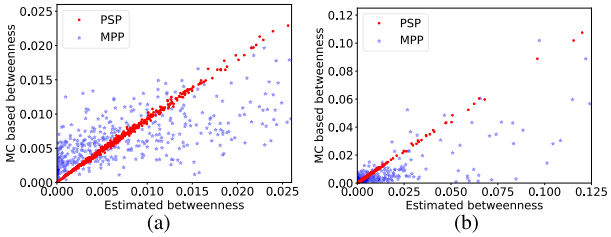

FIGURE 6. Scatter plots of the MC-based Betweenness versus estimated one. (a) ER model. (b) BA model.

Fig. 5 illustrates the results. Our method outperforms the MPP method in both accuracy and stability. The lower bounds of edge probabilities have little impact on the effectiveness of our method. However, the SCC of the MPP method decreases versus the increase of the lower bound. It is because the generated probabilistic graphs are becoming more deterministic with the increase of the lower bounds of edge probabilities. Thus, using the most probable paths to approximate the shortest paths results in more substantial estimation errors.

To gain an insight into the results, we randomly select one of the experiments and present the scatter plot of the MC-based betweenness versus the betweenness estimated by our method and the MPP method. Fig. 6 presents the results. We obtain similar results for other SCC experiments. We find that the results obtained by our method are linearly correlated with the MC-based betweenness for both network models. However, the MPP approach tends to under-estimate the betweenness of the nodes which are not essential and over-estimate the betweenness of the nodes which have relatively high MC-based betweenness. These findings further verify the effectiveness of our method.

4) IMPACTS OF NETWORK PROPERTIES

In this experiment, we investigate how the number of nodes impacts the effectiveness of our method. We keep the network density as a constant in the experiment. Fig. 7 illustrates how SCC changes versus the number of nodes. The results show that our method is more effective in estimating the betweenness centrality of nodes. Our method obtains much higher correlation coefficients than the MPP method. Additionally, our method is also more stable than the MPP method for both network models.

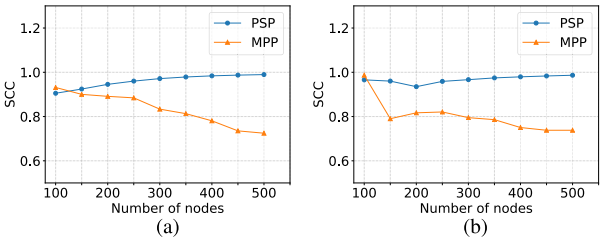

FIGURE 7. SCC versus the number of nodes. (a) ER model. (b) BA model.

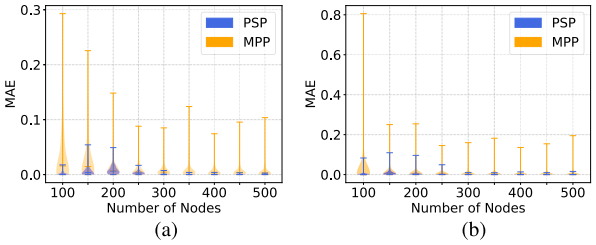

FIGURE 8. Violin plot of absolute errors versus the number of nodes. (a) ER model. (b) BA model.

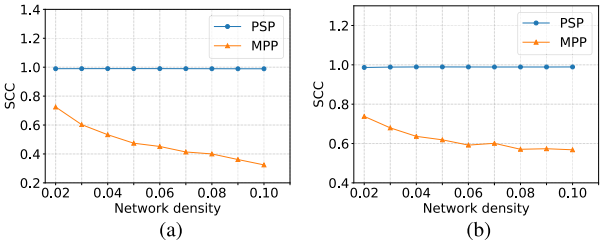

FIGURE 9. SCC versus the network densities. (a) ER model. (b) BA model.

To gain an insight into the performance of our method, we also calculate the absolute errors of our method and the MPP method. Fig. 8 presents the violin plot of the results. The MAE for both the ER and BA models are not sensitive to the change of the number of nodes. Moreover, the majority of the absolute errors are sufficiently small for both network models, further confirming the effectiveness of our method. Besides, the absolute errors of the MPP method are much larger than that of ours for both network models. The results also show that the MPP method has much more significant variances of absolute errors than ours, indicating the effectiveness of our algorithm in estimating the actual values of betweenness centrality.

We also examine the impacts of network densities on the effectiveness of our method. In the experiment, we keep the number of nodes as a constant. Fig. 9 plots the SCC versus the network density. The results show that the network density has little impacts on the results of our method. Besides, the accuracy of the MPP method decreases versus the increase of network density. It indicates that the most probable path is not a good substitution for the shortest path, especially when the probabilistic graphs are dense.

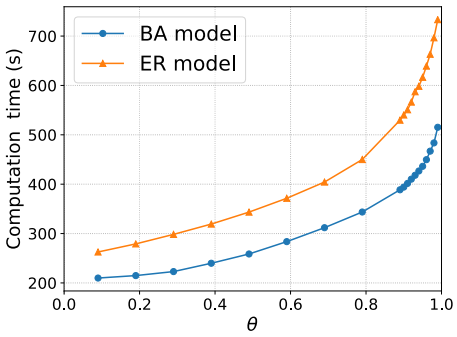

FIGURE 10. Efficiency evaluation of the parameter θ.

C. EFFICIENCY EVALUATION

In this experiment, we evaluate the efficiency of our method by examining the computation time used to calculate the betweenness centrality of nodes. The experiments are implemented in Python 2.7 and run on a server with 64-bit Ubuntu 16.04.5, Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 128.0GB RAM.

1) IMPACTS OF THE THRESHOLD θ

We first evaluate the impacts of the parameter θ on the efficiency of our method. The parameter θ is the connectivity threshold to indicate when the algorithm can terminate with acceptable accuracy. Although a more significant value of θ ensures a more substantial coverage of the all possible shortest paths between a node pair, it demands more computations. By adjusting the value of θ, we examine the total computation times used to explore the possible shortest paths between all node pairs of the network. Fig. 10 illustrates the computation times for different values of θ. The results show that the computation time gradually increases versus θ. When θ approaches to 1.0, indicating an exhaustive exploration effort of all possible shortest paths between a node pair, the computation time increases sharply. Given that the estimation accuracy maximizes at $\theta \approx 0.8$, we strongly suggest that $\theta = 0.8$ is a suitable choice to achieve high effectiveness and efficiency.

2) IMPACTS OF NETWORK DENSITY

Since the MC method may converge earlier than the theoretical number of samples, we define a new metric to indicate the convergence of the MC method. In each round of the MC method, we calculate the average betweenness centrality of all nodes. Then we calculate the differences of the average betweenness centrality between two consecutive epochs. If the differences of the average betweenness centrality are less than 10^{-6} within five consecutive rounds, we say that the MC simulation converges. Fig. 11 presents the convergence time of the MC method based on the ER and BA models, respectively. The red points mark the convergences based on our metric. The results show that the simulations reach convergence with much fewer samples than the theoretical number of samples.

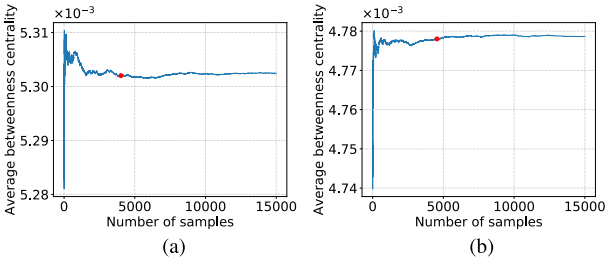

FIGURE 11. Convergence time of the MC simulations. (a) ER model. (b) BA model.

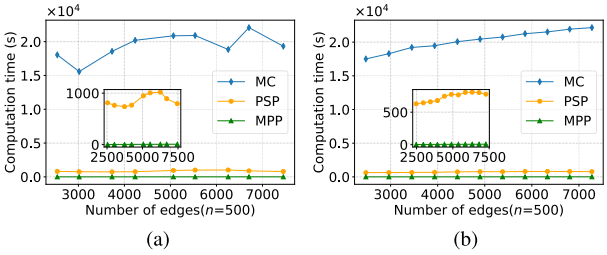

FIGURE 12. Computation time versus the number of edges. (a) ER model. (b) BA model.

Fig. 12 presents the computation time versus the network density. We find that the computation time used by the MC method increases linearly versus the number of edges approximately. It is because the MC method has to sample the probabilistic graph. Therefore, more edges indicate more trials to instantiate a realization of the probabilistic network. It is precisely why the computation time grows linearly versus the network density for the Monte Carlo method. Compared to the MC method, the MPP method and our method cost much less time to calculate the betweenness centrality. The inset figures present the details of the two methods. The results show that the computation time used by our method increases slightly versus the increase of network density. The MPP method costs the least computation time. It is because the MPP method only computes the probable paths. Our algorithm consumes more time than the MPP method. However, our method is more efficient than the MC approach while obtains comparable accuracy. Although the MPP method has much higher computational efficiency, it has more calculation errors in the possible world semantics.

As shown in the inset figure, there is a transition phase of the computation time for our method. Before the transition point, the computation time is almost constant as the network density increases. However, after the transition point, the computation time grows slight versus the network density. It can be explained as follows: when the density of a probabilistic network is lower than a threshold, the number of possible shortest paths between pairs of nodes is almost constant. However, when the density is higher than the threshold, the number of possible shortest paths increases if the network contains more edges. To verify this conjecture, we plot the number of possible shortest paths between node pairs versus

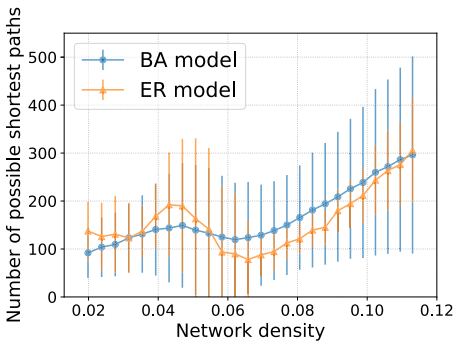

FIGURE 13. Number of possible shortest paths between node pairs versus network density.

TABLE 2. A summary of the number of edges.

	Max	Min	Mean	Std
Daily	40029	19771	29133.4	3635.8
Monthly	54402	38319	47452.0	5177.4

the network density in Fig. 13. The vertical lines illustrate the error bars at each data point. The average number of possible shortest paths displays a transition phase with the increase of network density. Besides, the number variance of possible shortest paths between two nodes becomes more significant when the probabilistic graphs become denser. We believe that the transition point is relevant to the distribution of edge probabilities, though the exact relationship between them is still an open problem.

VI. APPLICATION

In this section, we apply the proposed metrics to evaluate the importance of autonomous systems (ASes) on the Internet.

A. DATASET

We mainly focus on the core Internet composed of the top autonomous systems ranked by the CAIDA [14] since these ASes are more vital than others. The CAIDA ranks the ASes with four different metrics, namely the number of ASes in customer cone (#ASes), the number of IPv4 prefixes in customer cone (#Pref), the number of IPv4 address in customer cone (#Addr), and the AS transit degree (#Tran). We retrieve the top 1000 ASes for each metric and get 1996 distinct ASes. We then build the topology of these ASes using the *BGPStream* platform, which is an open-source software framework for the analysis of both historical and real-time Border Gateway Protocol (BGP) measurement data [15]. To capture the dynamics of the Internet on different timescales, we construct two types of deterministic networks from 2015 to 2016 by day and by month, respectively. We obtained 731 daily and 24 monthly deterministic networks. A summary of the number of edges is presented in Table 2.

In order to examine the dynamic nature of the Internet, we calculate the edge overlapping ratio between consecutive

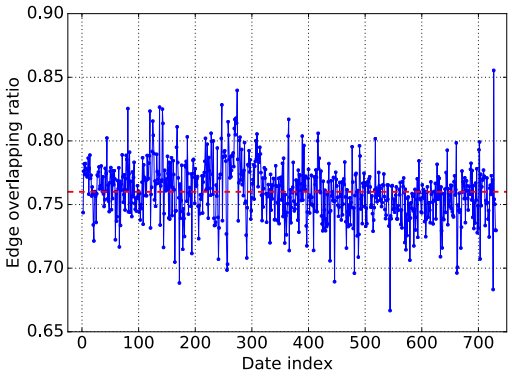

FIGURE 14. Edge overlapping ratios between consecutive daily networks.

deterministic networks as:

$$r = \frac{|E_t \cap E_{t+1}|}{|E_t \cup E_{t+1}|}$$

where E_t is the edge set of the network at time t, and $|\cdot|$ represents the cardinal of a set. Fig. 14 shows the overlapping edge ratio of the daily networks. The dashed horizontal line illustrates the mean overlapping ratio which is about 0.76. We find that nearly a quarter of the edges change over time, indicating that a proper model is desired to capture the dynamic nature of the Internet. As demonstrated in previous sections, the probabilistic graph model is a suitable choice in modeling dynamic networks. In this following experiments, we build probabilistic graphs based on the above obtained deterministic networks of the Internet.

B. EFFECTIVENESS VALIDATION

We evaluate the importance of obtained nodes in two approaches. The first is to compare the ranks of ASes based on our proposed probabilistic betweenness centrality with the AS ranks ordered by the CAIDA. Since the ranks of ASes ordered by the CAIDA are based on the dataset up to 1st June 2016, we build the probabilistic graph of the Internet based on the three monthly deterministic networks from March to May 2016. We explicitly remove the stub nodes in the built probabilistic graph for computation efficiency. The final probabilistic graph contains 1893 nodes and 54241 edges.

Table 3 lists the top 10 ASes for different metrics. *Btwn* indicates the probabilistic betweenness centrality of the AS calculated based on the built probabilistic graph. *Deg* represents the degree of the AS in the probabilistic graph. *#ASes* is the number of ASes in customer cone. *#Addr* is the number of addresses in customer cone. *#Pref* is the number of prefixes in customer cone. *#Trans* is the transit degree of the AS. The last four metrics are used by the CAIDA to rank the ASes. Comparing the top 10 ASes obtained by different metrics, we find that the proposed probabilistic betweenness centrality can retrieve most of the substantial ASes promoted by the CAIDA metrics. Besides, the probabilistic betweenness centrality also promotes ASes which might be underestimated by other metrics. We highlight these ASes with bold in the second column in Table 3. AS9498 is owned by Bharti Airtel Limited [16] which is a leading global telecommunications company with operations in 20 countries across Asia and Africa. AS24482 is operated by SG.GS which is a Singapore Network Provider [17]. AS12989 is registered by ORG-EIS3-RIPE, owned by a Netherlands Internet provider, Eweka Internet Services B.V. [18]. All of these ASes are Internet Service Providers that play essential roles in the global communication of the Internet. It is worth noting that most of the top 10 ASes promoted by the *Deg* metric are also similar to those promoted by other metrics. It indicates that most of the critical ASes have many connections with other ASes. It coincides with the intuition that important ASes tend to be allocated with more resources.

TABLE 3. Top 10 ASes for each metric.

Ranks	Btwn	Deg	#ASes	#Addr	#Pref	#Trans
1	6939	6939	3356	1	3356	6939
2	174	174	174	1299	1299	174
3	1299	24482	1299	2914	2914	3356
4	3257	1299	2914	174	174	3549
5	3549	3257	3257	3257	3257	7018
6	3356	36236	6762	6453	6453	20485
7	**9498**	9498	6453	701	6762	8220
8	**24482**	12989	6939	1239	6939	43531
9	2914	3356	9002	6762	2828	4323
10	**12989**	43531	1273	6939	3491	209

To further validate our speculation, we also calculate the Spearman correlations of the top 100 ASes across these metrics. The Spearman correlation is defined as:

$$\rho = 1 - \frac{6\sum(r_X^i - r_Y^i)}{n(n^2-2)}$$

where r_X^i and r_Y^i is the rank of i-th object according to X and Y, respectively; and n is the number of objects. Table 4 shows the results. The results demonstrate that the metric of probabilistic betweenness centrality has high correlations with the metrics used by the CAIDA. It indicates that the metric of probabilistic betweenness centrality indeed captures the importance of the ASes on the Internet. The highest correlation is 0.7347 with the metric of AS transit degree. This fact shows that the metric of probabilistic betweenness centrality does not depend on any of the existing metrics. The moderate correlations between the metrics suggest that our proposed probabilistic betweenness centrality reconciles the conflicts of the current metrics with a global perspective. Moreover, the metric of node degree also has high correlations with the CAIDA metrics. It is because the probabilistic graph is built with a period of three months. Therefore, active ASes tend to have relatively high degrees in the graph. Those results show that both probabilistic betweenness centrality and node degree are useful metrics to evaluate the importance of an AS on the Internet.

TABLE 4. Spearman correlations.

	Btw	Deg	#ASes	#Addr	#Pref	#Trans
Btw	1.0	0.8525	0.6941	0.6004	0.5670	0.7347
Deg	-	1.0	0.6636	0.6949	0.7065	0.7511
#ASes	-	-	1.0	0.8618	0.7582	0.7732
#Addr	-	-	-	1.0	0.8115	0.6658
#Pref	-	-	-	-	1.0	0.6953
#Trans	-	-	-	-	-	1.0

The second approach is to evaluate the disintegration of the network via the removal of nodes [19] in an order sorted by different metrics. The disintegration of a deterministic graph G is defined as:

$$\eta = \frac{1}{n} \sum_{s<s_{\max}} n_s s^2$$

where n_s is the number of connected components containing s nodes; and n is the number of nodes in G. The disintegration measure the average size of the connected components except for the giant one after the removal of a node. According to the percolation theory [20], [21], the divergence point of the disintegration indicates a phase transition of network collapse. A useful metric should lead to a relatively earlier collapse of the network if we remove the nodes continually in the order of the metric. In the experiment, we remove the nodes continually according to their ranks ordered by different metrics and then calculate the disintegration of the networks after the removal of each node. As for probabilistic graphs, we calculate the expected disintegration after the removal of a node. Explicitly, we instantiate N realizations of the probabilistic graph and approximate the expected disintegration with the mean disintegration of these instantiated networks. We set $N = 100$ in the experiments. Fig. 15 presents the results. Besides the four CAIDA metrics, we also compare the effectiveness of probabilistic betweenness centrality with the metric of node degree. We find that node removals based on the four CAIDA metrics do not result in any apparent network collapses. In contrast, both the metrics of probabilistic betweenness centrality and node degree successfully result in network collapses with the continual removals of nodes. However, our metric achieves an earlier collapse (around at the 550th node) than that of node degree (about at the 950th node). It demonstrates that the proposed probabilistic betweenness centrality is more effective in assessing the vulnerability of the Internet.

Conclusion: The proposed metric of probabilistic betweenness centrality can promote these important ASes ranked high by the four CAIDA metrics. Besides, the metric also results in the earliest collapse of the network compared to other metrics. These results validate the effectiveness of the metric in evaluating the importance of ASes on the Internet.

VII. RELATED WORK

Link uncertainty arises due to various reasons from the dynamics of networks to data processing. Frank Harary

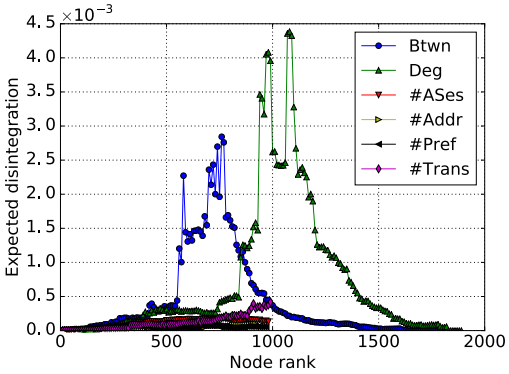

FIGURE 15. Effectiveness evaluation of proposed metrics based on expected disintegration.

coined the notion of probabilistic graphs in 1969 [22]. Since then, the probabilistic graph is widely used to characterize the link uncertainty of graphs in many domains from biological networks [23], [24], social networks [25], [26], and communication networks [27], [28], to database queries [29]–[31]. Due to the primary importance and great theoretical value of probabilistic graphs, there are many studies relevant to this work.

Several studies focus on measuring the criticality of nodes and edges in probabilistic graphs. Frank first studied the problem of finding shortest-path probability distributions in graphs [22]. Sigal *et al.* [32] studied the shortest routing problem in probabilistic graphs. Ji [33] studied the shortest path problem with stochastic link length. They initially proposed the concepts of the expected shortest path and the most probable shortest path based on different decision criteria. Potamias *et al.* [2] studied the problem of the k-nearest neighbor queries in probabilistic graphs. They proposed novel distance functions by extending the concept of the shortest path in deterministic graphs. The Monte Carlo methods are used to determine the shortest path probabilities between the edges. Hua and Pei [34] extend this to find the shortest weighted paths most likely to complete within a specific time constraint. Pfeiffer and Neville [1] developed measures of path length, betweenness centrality, and clustering coefficient in probabilistic graphs based on sampling and probabilistic paths. In this paper, we complement the study of the shortest path in probabilistic graphs by firstly proposing the concept of possible shortest paths. Different from previously studied where the length of the shortest path between two nodes in a graph (either deterministic or probabilistic) is deterministic, our definition of possible shortest paths allows these paths have different lengths and associates a probability to each of these paths. Besides, we further formally define the measure of probabilistic betweenness centrality based on the concept of possible shortest paths, which fill the gap between probabilistic and deterministic graphs.

There have been some studies addressing the vulnerability assessment of probabilistic networks. Aggarwal *et al.* [35]

studied the reliability of communication systems where links, as well as nodes, have a certain probability of failure. Colbourn [36] formulated the resilience of a network as the expected number of node pairs which can communicate in the network. They examined the analysis and synthesis problems for resilience in a setting where links fail independently with known probabilities. Karger [37] introduced the first fully polynomial randomized approximation scheme for the all-terminal-reliability problem where each of the edges causes failures independently with some given probability. Neumayer and Modiano [38] proposed some network performance metrics to evaluate the average two-terminal reliability in polynomial time under independent random line-cuts. Agarwal *et al.* [39] developed a unified framework to evaluate the vulnerability of communication networks when the disruptive event has a probabilistic nature, defined by an arbitrary probability density function. They employed computational geometric tools to achieve efficient algorithms to identify vulnerable points within the network under various metrics. Dinh and Thai [3] adopted the expected pairwise connectivity as a measure to quantify global connectivity in probabilistic graphs and utilized it to formulate vulnerability assessment as a stochastic optimization problem. Their method avoids considering all possible realizations of probabilistic graphs and can efficiently estimate the expected pairwise connectivity with any desired accuracy. However, these methods are still computation-intensive and are not capable of the evaluation of large-scale networks.

In this paper, we formally define the probabilistic betweenness centrality based on the concept of possible shortest paths. Moreover, we propose a heuristic algorithm to find possible shortest paths between two nodes and thus calculate the betweenness centrality efficiently. The extensive experiments demonstrate the effectiveness and efficiency of our method in assessing the centrality of probabilistic graphs.

VIII. CONCLUSION

In this paper, we propose the concept of possible shortest paths in probabilistic graphs. Then we formally define the betweenness centrality of probabilistic graphs. The definition can be generalized to the case of deterministic ones. To solve the efficiency issue, we develop a heuristic algorithm to explore the possible shortest paths and implement an efficient algorithm to calculate the betweenness centrality. The algorithms avoid the sampling process and thus significantly improve the computational efficiency. We validate the effectiveness and efficiency of our method with extensive experiments. The experimental results show that our approach well captures the centrality of probabilistic graphs in the possible-world semantics. In future work, we would like to investigate the applications of probabilistic graphs on the evaluation of dynamic graphs such as evolving social networks.

APPENDIX. ALGORITHM ERROR ANALYSIS

Theorem 2: The betweenness centrality error of Algorithm 2 caused by the exception presented in Fig. 2 is up-bounded by a small value of 0.083.

Proof: First, we prove that the case presented in Fig. 2 leads to the maximum errors amongst all exceptional cases. According to Eqn (8), the values of betweenness centrality are inversely proportional to and dominated by the square of the number of nodes. Thus, the errors are also inversely proportional to and dominated by the square of the number of nodes. That is, networks with fewer nodes have more substantial calculation errors. The graph presented in Fig. 2 has the fewest nodes amongst all exceptional cases. Therefore, the errors of our algorithm are up-bounded by the maximum error of the graph presented in Fig. 2.

Second, we calculate the maximum error of the graph presented in Fig. 2. It is easy to see that the error is maximized when $p_3 = p_4 = 1.0$. In the following, we compute the maximum error under the condition that $p_3 = p_4 = 1.0$ and $p_1 < p_2$. In this case, all possible shortest paths sets are:

$$\begin{aligned}
\mathcal{S}_{12} &= \left\{S_1^{12} = [v_1, v_2]\right\},\\
\mathcal{S}_{13} &= \left\{S_1^{13} = [v_1, v_2, v_3], S_2^{13} = [v_1, v_2, v_4, v_3]\right\}\\
\mathcal{S}_{14} &= \left\{S_1^{14} = [v_1, v_2, v_4]\right\}\\
\mathcal{S}_{23} &= \left\{S_1^{23} = [v_2, v_3], S_2^{23} = [v_2, v_4, v_3]\right\}\\
\mathcal{S}_{24} &= \left\{S_1^{24} = [v_2, v_4]\right\}\\
\mathcal{S}_{34} &= \left\{S_1^{34} = [v_3, v_4]\right\}
\end{aligned}$$

According to the Equation 8, the Betweenness Centrality of v_1 can be calculated theoretically by:

$$\begin{aligned}
c_B(v_1) &= \frac{2}{(4-1)(4-2)}\left[\frac{\widehat{\Pr}(S_1^{12})}{\widehat{\Pr}(S_1^{12})}\varphi_{12}\right.\\
&\quad\left. + \frac{\widehat{\Pr}(S_1^{13}) + \widehat{\Pr}(S_2^{13})}{\widehat{\Pr}(S_1^{13}) + \widehat{\Pr}(S_2^{13})}\varphi_{13} + \frac{\widehat{\Pr}(S_1^{12})}{\widehat{\Pr}(S_1^{12})}\varphi_{14}\right]\\
&= \frac{1}{3}(\varphi_{12} + \varphi_{13} + \varphi_{14})
\end{aligned}$$

However, Algorithm 1 fails to find the path S_2^{13}. Thus the connectivity of nodes v_1 and v_3 is different from the theoretical value, and the Betweenness Centrality of v_1 is computed as

$$\begin{aligned}
c'_B(v_1) &= \frac{2}{(4-1)(4-2)}\left[\frac{\widehat{\Pr}(S_1^{12})}{\widehat{\Pr}(S_1^{12})} * \varphi_{12}\right.\\
&\quad\left. + \frac{\widehat{\Pr}(S_1^{13})}{\widehat{\Pr}(S_1^{13})} * \varphi'_{13} + \frac{\widehat{\Pr}(S_1^{12})}{\widehat{\Pr}(S_1^{12})} * \varphi_{14}\right]\\
&= \frac{1}{3}(\varphi_{12} + \varphi'_{13} + \varphi_{14})
\end{aligned}$$

Then the absolute error of v_1 can be calculated by:

$$\Delta_{v_1} = \frac{1}{3}|\varphi_{13} - \varphi'_{13}| \tag{12}$$

We can easily find out that:

$$\varphi_{13} = 1 - (1 - \widehat{P}(S_1^{13}))(1 - \widehat{\Pr}(S_2^{13})) = p_1 + p_1^2 p_2 - p_1^2 p_2^2$$
$$\varphi'_{13} = 1 - (1 - \widehat{\Pr}(S_1^{13})) = \widehat{\Pr}(S_1^{13}) = p_1 p_2$$

while $\widehat{P}(S_1^{13}) = p_1 p_2$ and $\widehat{P}(S_2^{13}) = p_1(1 - p_2)$. On the basis of the prerequisite that $p_1 < p_2 \leq 1$, we can figure out that:

$$\begin{aligned}
\Delta_{v_1} &= \frac{1}{3}[(p_1 + p_1^2 p_2 - p_1^2 p_2^2) - p_1 p_2] \\
&= \frac{1}{3}[p_1(1 - p_1 p_2) - p_1 p_2(1 - p_1 p_2)] \\
&= \frac{1}{3}[p_1(1 - p_2)(1 - p_1 p_2)] < \frac{1}{3}[p_2(1 - p_2)(1 - p_1 p_2)] \\
&< \frac{1}{3}[p_2(1 - p_2)] = \frac{1}{3}[-(p_2 - \frac{1}{2})^2 + \frac{1}{4}] \\
&< \frac{1}{3} * \frac{1}{4} = \frac{1}{12} \approx 0.083
\end{aligned}$$

Analogously, we can compute the absolute error for other vertexes as:

$$\begin{aligned}
c_B(v_2) &= \frac{1}{3}(\varphi_{12} + \varphi_{13} + \varphi_{23} + \varphi_{24}) \\
c'_B(v_2) &= \frac{1}{3}(\varphi_{12} + \varphi'_{13} + \varphi_{23} + \varphi_{24}) \\
c_B(v_3) &= \frac{1}{3}(\varphi_{13} + \varphi_{23} + \varphi_{34}) \\
c'_B(v_3) &= \frac{1}{3}(\varphi'_{13} + \varphi_{23} + \varphi_{34}) \\
c_B(v_4) &= \frac{1}{3}(\frac{\widehat{\Pr}(S_1^{13})}{\widehat{\Pr}(S_1^{13}) + \widehat{\Pr}(S_1^{23})}\varphi_{13} + \varphi_{14} + \varphi_{24} + \varphi_{34}) \\
&= \frac{1}{3}[(1 - p_2)\varphi_{13} + \varphi_{14} + \varphi_{24} + \varphi_{34}] \\
c'_B(v_4) &= \frac{1}{3}(\varphi_{14} + \varphi_{24} + \varphi_{34})
\end{aligned}$$

Apparently, the absolute errors of v_2 and v_3 equal to that of v_1:

$$\Delta_{v_2} = \Delta_{v_3} = \Delta_{v_1} = \frac{1}{3}|\varphi_{13} - \varphi'_{13}| \tag{13}$$

while absolute error of Δ_{v_4} is

$$\begin{aligned}
\Delta_{v_4} &= \frac{1}{3}(1 - p_2)\varphi_{13} \\
&= \frac{1}{3}(p_1 - p_1^2 p_2 + 2p_1^2 p_2^2 - p_1 p_2 - p_1^2 p_2^3)
\end{aligned} \tag{14}$$

Since $p_1 - p_1^2 p_2 + 2p_1^2 p_2^2 - p_1 p_2 - p_1^2 p_2^3 = p_1(1 - p_2)(1 - p_1 p_2 + p_1 p_2^2) < p_1(1 - p_2)$, we can safely conclude that

$$\Delta_{v_4} < \frac{1}{3} * \frac{1}{4} = \frac{1}{12} \approx 0.083 \tag{15}$$

According to equation(15) and (13), the absolute errors of our algorithms is bounded to a maximum value of 0.083, which has a limited impact on the result. ■

REFERENCES

[1] J. J. Pfeiffer, III, and J. Neville, "Methods to determine node centrality and clustering in graphs with uncertain structure," in *Proc. ICWSM*, 2011, pp. 590–593. [Online]. Available: http://arxiv.org/abs/1104.0319

[2] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios, "K-nearest neighbors in uncertain graphs," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 997–1008, 2010.

[3] T. N. Dinh and M. T. Thai, "Assessing attack vulnerability in networks with uncertainty," in *Proc. INFOCOM*, vol. 26, Apr./May 2015, pp. 2380–2388.

[4] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[5] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the North American power grid," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, 2004, Art. no. 025103.

[6] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 1, 2002, Art. no. 056109.

[7] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas, "Characterization of complex networks: A survey of measurements," *Adv. Phys.*, vol. 56, no. 1, pp. 167–242, 2007.

[8] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2193–2200, 2009.

[9] G. Kollios, M. Potamias, and E. Terzi, "Clustering large probabilistic graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 325–336, Feb. 2013.

[10] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM J. Comput.*, vol. 8, no. 3, pp. 410–421, Aug. 1979.

[11] M. Riondato and E. M. Kornaropoulos, "Fast approximation of betweenness centrality through sampling," *Data Mining Knowl. Discovery*, vol. 30, no. 2, pp. 438–475, 2016.

[12] Z. Liu, Y.-C. Lai, and N. Ye, "Propagation and immunization of infection on general networks with both homogeneous and heterogeneous components," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 67, no. 3, 2003, Art. no. 031911.

[13] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.

[14] CAIDA. (2017). *As Rank*. Accessed: Mar. 6, 2017. [Online]. Available: http://as-rank.caida.org/

[15] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti, "BGPStream: A software framework for live and historical BGP data analysis," in *Proc. ACM Internet Meas. Conf.*, 2016, pp. 429–444.

[16] Airtel. (2017). *Artel*. Accessed: Mar. 6, 2017. [Online]. Available: http://www.airtel.in/

[17] SG.GS. (2017). *SG.GS: Singapore Network Provider*. Accessed: Mar. 6, 2017. [Online]. Available: http://www.sg.gs/

[18] EWEKA. (2017). *Usenet Provider*. Accessed: Mar. 6, 2017. [Online]. Available: https://www.eweka.nl/

[19] J.-P. Onnela *et al.*, "Structure and tie strengths in mobile communication networks," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 18, pp. 7332–7336, 2007.

[20] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*. Boca Raton, FL, USA: CRC Press, 1994.

[21] A. Bunde and S. Havlin, *Fractals and Disordered Systems*. Berlin, Germany: Springer, 2012.

[22] H. Frank, "Shortest paths in probabilistic graphs," *Oper. Res.*, vol. 17, no. 4, pp. 583–599, Aug. 1969.

[23] S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth, "Predicting protein complex membership using probabilistic network reliability," *Genome Res.*, vol. 14, no. 6, pp. 1170–1175, 2004.

[24] P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen, "Link discovery in graphs derived from biological databases," in *Proc. Int. Workshop Data Integr. Life Sci.* Berlin, Germany: Springer, 2006, pp. 35–49.

[25] E. Adar and C. Re, "Managing uncertainty in social networks," *IEEE Data Eng. Bull.*, vol. 30, no. 2, pp. 15–22, Jul. 2007.

[26] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.

[27] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 133–144, 2005.

[28] J. Ghosh, H. Q. Ngo, S. Yoon, and C. Qiao, "On a routing problem within probabilistic graphs and its application to intermittently connected networks," in *Proc. INFOCOM*, May 2007, pp. 1721–1729.
[29] L. Antova, T. Jansen, C. Koch, and D. Olteanu, "Fast and simple relational processing of uncertain data," in *Proc. IEEE 24th Int. Conf. Data Eng. (ICDE)*, Apr. 2008, pp. 983–992.
[30] N. Dalvi and D. Suciu, "Efficient query evaluation on probabilistic databases," *Int. J. Very Large Data Bases*, vol. 16, no. 4, pp. 523–544, 2007.
[31] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Zuefle, "Probabilistic frequent itemset mining in uncertain databases," in *Proc. SIGKDD*, 2009, pp. 119–128.
[32] C. E. Sigal, A. A. B. Pritsker, and J. J. Solberg, "The stochastic shortest route problem," *Oper. Res.*, vol. 28, no. 5, pp. 1122–1129, 1980.
[33] X. Ji, "Models and algorithm for stochastic shortest path problem," *Appl. Math. Comput.*, vol. 170, no. 1, pp. 503–514, 2005.
[34] M. Hua and J. Pei, "Probabilistic path queries in road networks: Traffic uncertainty aware path selection," in *Proc. 13th Int. Conf. Extending Database Technol.*, 2010, pp. 347–358.
[35] K. Aggarwal, J. Gupta, and K. Misra, "A simple method for reliability evaluation of a communication system," *IEEE Trans. Commun.*, vol. 23, no. 5, pp. 563–566, May 1975.
[36] C. J. Colbourn, "Analysis and synthesis problems for network resilience," *Math. Comput. Model.*, vol. 17, no. 11, pp. 43–48, 1993.
[37] D. R. Karger, "A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem," *SIAM Rev.*, vol. 43, no. 3, pp. 499–522, 2001.
[38] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *Proc. INFOCOM*, Mar. 2010, pp. 1–9.
[39] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, and G. Zussman, "The Resilience of WDM networks to probabilistic geographical failures," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1525–1538, Oct. 2013.

CHENXU WANG received the B.S. degree in communication engineering and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, in 2009 and 2015, respectively, where he is currently an Assistant Professor with the School of Software Engineering. He was a Postdoctoral Research Fellow of the Hong Kong Polytechnic University. His current research interests include complex network analysis, network security, online social network analysis, and information diffusion.

ZIYUAN LIN received the B.S. degree from Xi'an Jiaotong University, in 2019, where he is currently a Research Assistant with the MoE Key Lab of Intelligent Networks and Network Security. His current research interests include data mining and complex network analysis.