MULTI-VARIABLE LINEAR REGRESSION-BASED PREDICTION OF A COMPUTATIONALLY-HEAVY LINK STABILITY METRIC FOR MOBILE SENSOR NETWORKS

Natarajan Meghanathan

Jackson State University, 1400 Lynch St, Jackson, MS, USA

ABSTRACT

Until now, we were determining stable data gathering (DG) trees for mobile sensor networks (MSNs) using a link stability metric (computationally-light or computationally-heavy) that is directly computed on the egocentric edge network. Among such DG trees, the BPI' (complement of bipartivity index)-based DG trees were observed to be the most stable, but the BPI' metric is also computationally-heavy. Hence, we seek to build a multi-variable linear regression model to predict the BPI' values for the egocentric networks of edges using three computationally-light metrics (neighborhood overlap: NOVER, one-hop two-hop neighborhood: OTH, and normalized neighbor degree: NND) that are also computed on the egocentric edge networks. The training and testing are conducted as part of a single simulation run (i.e., in-situ). The training dataset comprises of the BPI', NOVER, OTH and NND values of randomly sampled egocentric edge networks during the first phase of the simulation (1/5th of the total simulation time). We observe the R-square values for the prediction to be above 0.85 for both low density and high density networks. We also observe the lifetimes of the predicted BPI'-based DG trees to be 87-92% and 55-75% of the actual BPI'-based DG trees for low-moderate and moderate-high density networks respectively.

KEYWORDS

Multi-variable Regression, Bipartivity Index, Computationally-Light, Computationally-Heavy, Mobile Sensor Networks, Data Gathering Tree

1. INTRODUCTION

Mobile sensor networks (MSNs) are characteristic of dynamically changing topology in which the sensor nodes could move independent of each other. The data gathering (DG) tree (network-wide spanning tree) was considered the most energy efficient communication structure for wireless sensor networks [1]. But, due to the dynamic nature of MSNs, the DG trees need to be frequently reconfigured. The problem of determining stable DG trees in MSNs has been studied for the last few years. The initial approach was to adapt the solutions for stable routing in mobile ad hoc networks (MANETs [2]) and under such an approach, the predicted link expiration time (LET [3])-based DG trees [4] were observed to be the most stable for MSNs. However, the LET and other MANET-based techniques require the sensor nodes to be location and mobility-aware.

In [5], a suite of location and mobility-independent metrics (adapted from Network Science) were proposed for quantifying link stability in the form of link stability scores that could be used as edge weights to determine stable DG trees. The hypothesis behind the use of such Network

Natarajan Meghanathan et al. (Eds) : CSITEC, CMCA, SP, NECOM, ADCO - 2019
pp. 29-38, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91103

Science-based metrics is that edges whose end vertices share a significant fraction of their neighbors are more likely to be stable (the end vertices of such edges are expected to be also closer to each other and hence would stay as neighbors for a longer time). Hence, Network Science metrics that could quantify the extent of shared neighborhood between the end vertices of an edge were considered for adaptation to quantify link stability; the most promising among these Network Science metrics was the Bipartivity Index (BPI) metric [6]. DG trees that were determined by using the complement of BPI (i.e., BPI' = 1 - BPI) were observed to be the most stable among the DG trees determined using the Network Science metrics. The BPI'-based DG trees were also observed to be significantly more stable than the LET-based DG trees.

The BPI'-based link stability score for an edge is computed on the egocentric network of the edge. The egocentric network of an edge comprises of the end vertices of the edge and their neighbors as *vertices* and the edges incident on the end vertices as the *edges*. The primary weakness of the BPI' metric is that it is computationally-heavy and would take a significantly longer time to be computed for every edge in the network whenever a DG tree is required to be constructed. Hence, the motivation for further research in this direction is to develop computationally-light metrics that could be used as alternatives to quantify the extent of shared neighborhood (and in turn link stability). The Neighborhood Overlap (NOVER) metric [7] is a computationally-light metric that was also proposed for use in [5] (along with BPI') to quantify link stability. The lifetimes of the NOVER-DG trees [5, 8] were the second largest (i.e., next to the lifetimes of the BPI'-DG trees) for several scenarios of network density and node mobility.

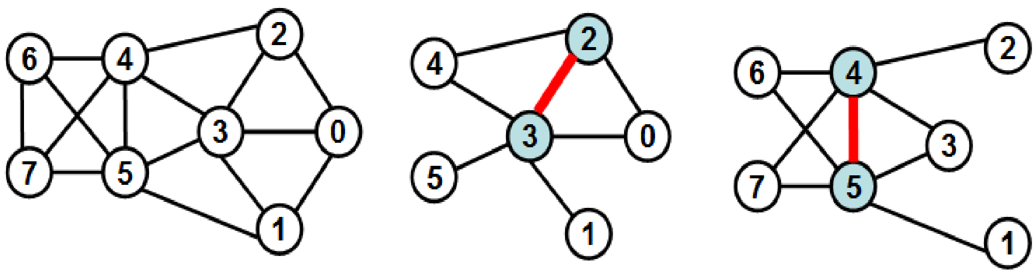
In this paper, we take a different approach compared to what has been done so far in the literature. Rather than computing the actual computationally-heavy metric (such as BPI') or computationally-light metric (such as NOVER) to quantify link stability and determining DG trees using these metric scores as edge weights, we seek to explore the development of one or more computationally-light metrics (we actually propose two such metrics in this paper) that could be used along with NOVER to predict the computationally-heavy BPI' metric and in turn use such predicted BPI' metric values as edge weights to determine DG trees. In this pursuit, we first propose two computationally-light metrics (Normalized Neighbor Degree: NND and One Hop Two Hop Neighbors: OTH) and then use the trio (NOVER, OTH and NND) as the independent variables to predict (model) BPI' using multi-variable regression. We observe the lifetimes of the predicted BPI'-based DG trees to be very close to that of the actual BPI'-based DG trees in networks of low to moderate density and networks of moderate-high mobility. However, as node density becomes high (and/or node mobility is low), the accuracy of the prediction drops and the lifetimes of the predicted BPI'-based DG trees is only about half of the lifetimes of the actual BPI'-based DG trees.

The rest of the paper is organized as follows: In Section 2, we propose the two computationally-light metrics NND and OTH as well as review the NOVER and BPI' metrics. We also illustrate their computation on the egocentric network of an edge using an example graph as well as analyze the rank-based correlation between any two of the four metrics. In Section 3, we describe our procedure for performing multi-variable regression in an in-situ simulation setting. In Section 4, we present the simulation results comparing the lifetimes of the DG trees determined using the actual vs. predicted BPI' values as well as the computation times of the procedures to determine the actual BPI' and the predicted BPI' values. Section 5 summarizes the contributions of this work and concludes the paper. Throughout the paper, the terms 'node' and 'vertex', 'link' and 'edge', 'network' and 'graph' are used interchangeably. They mean the same.

2. Link Stability Metrics

Our hypothesis for this research is that edges whose end vertices are closer to each other are more likely to be stable and the end vertices of such edges are more likely to share a significant fraction of their neighborhood. Hence, if we could quantify the extent of shared neighborhood between the end vertices of the edges using the graph theoretic metrics (adapted from Network Science) that could be locally computed on the egocentric networks of the edges, we could use these metric values as edge weights and expect to determine stable data gathering (DG) trees. This approach was first successfully tried in [5] wherein, we identified two Network Science metrics (Bipartivity Index: BPI and Neighborhood Overlap: NOVER) that could be used to quantify the extent of shared neighborhood in mobile sensor networks and the metric values (BPI' = 1 - BPI and NOVER) were used as edge weights to determine DG trees that are significantly more stable than that of the predicted link expiration time (LET)-based DG trees which was until then considered the best approach to determine stable DG trees in MSNs.

As motivated in Section 1, the BPI' metric is computationally-heavy. Hence, in this section, we propose two computationally-light metrics (NND and OTH) that along with NOVER could be used to predict the BPI' value for the egocentric network of an edge. From notation point of view, let $N(u)$ represent the set of neighbors of a vertex u and let $Ego_N(u, v)$ represent the set of vertices (vertices u, v and their neighbors, with each vertex represented exactly once) constituting the egocentric network of an edge (u, v). Figure 1 presents an example graph (that is also used as a running example in this section) and the egocentric networks for two of its edges (2, 3) and (4, 5).



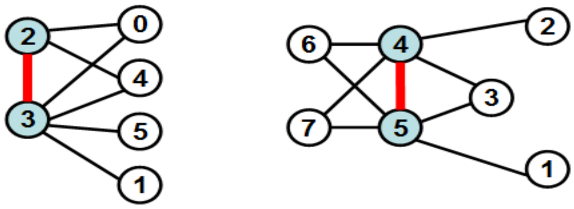
Graph | Egocentric Network for Edge (2, 3) | Egocentric Network for Edge (4, 5)

Figure 1. Example Graph and Egocentric Networks for some Sample Edges

2.1. Normalized Neighbor Degree (NND)

The NND metric (see formulation 1 below; k_i is the degree of node i) for an edge (u, v) is the ratio of the square root of the sum of the squares of the degrees of the neighbor nodes and the number of neighbor nodes in the egocentric network of the edge (u, v). The neighbor nodes of the end vertices u and v in the egocentric network of edge (u, v) could have a degree of either 1 or 2. If a neighbor node has degree 2, it implies the node is a neighbor of both u and v and is part of the shared neighborhood. If a neighbor node has degree 1, then the node is not part of the shared neighborhood. Hence, the larger the number of neighbor nodes with degree 2, the larger the extent of shared neighborhood. Figure 2 presents the computation of the NND for two edges (2, 3) and (4, 5) in the example graph of Figure 1.

$$NND(u,v) = \frac{\sqrt{\sum_{i \in Ego_N(u,v)-\{u,v\}} k_i^2}}{|Ego_N(u,v) - \{u,v\}|} \quad \text{.......} \ (1)$$



$$\frac{\sqrt{2^2+2^2+1^2+1^2}}{(6-2)} \qquad \frac{\sqrt{2^2+2^2+2^2+1^2+1^2}}{(7-2)}$$

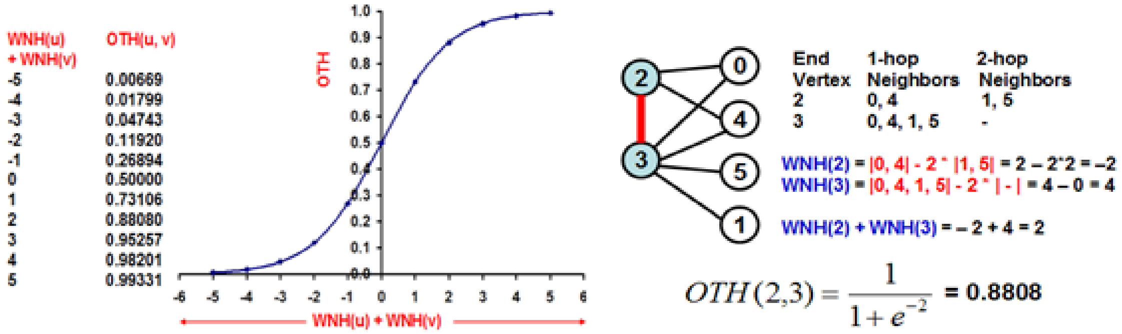
$$NND(2,3) = \mathbf{0.7905} \qquad NND(4,5) = \mathbf{0.7483}$$

Figure 2. Examples to Compute the Normalized Neighbor Degree (NND) for an Edge

2.2. One Hop Two Hop (OTH) Neighborhood

The OTH metric for an edge (*u*, *v*) is computed on the basis of the number of one hop and two hop neighbors of the end vertices *u* and *v* in the egocentric network of the edge. From the point of the end vertices *u* and *v*, it is desirable that all the vertices in $Ego_N(u, v)$ be their one hop neighbors. However, if a vertex in $Ego_N(u, v)$ is not present in the neighborhood of either *u* or *v*, then it becomes a two hop neighbor of the corresponding end vertex. The presence of two hop neighbors in the egocentric network of an edge (*u*, *v*) decreases the extent of shared neighborhood; on the other hand, the presence of one hop neighbors increases the extent of shared neighborhood. We propose an auxiliary metric called the weighted neighborhood hop count (WNH) as a weighted sum of the one hop and two hop neighbors of the end vertices *u* and *v* (see formulation 2). As seen in the WNH formulation for an end vertex, we complement the end vertex for having one hop neighbors and penalize for having two hop neighbors (by a factor of 2). In other words, the presence of two hop neighbors is modeled to weaken the stability of the neighborhood and negate the advantage that comes with the presence of one hop (shared) neighbors.

The value for WNH for an end vertex could be either negative or positive and its upper bound and lower bound depend on the number of vertices in the network. Also, to be noted is that though the WNH formulation penalizes a vertex for having two hop neighbors, the formulation is still linear. For two egocentric edge networks with the same number of edges, we anticipate the egocentric edge network with a relatively larger number of one-hop neighbors to be significantly stable compared to the egocentric edge network with a relatively larger number of two-hop neighbors.



Sigmoid Nature of the OTH Formulation   Example to Find the OTH Score for an Edge

Figure 3. OTH Formulation and a Sample Computation

The OTH for an edge (*u*, *v*) is computed using the WNH scores of the end vertices *u* and *v* (as shown in formulation 3). We opine the sigmoid function used in the formulation of the cost function for logistic regression [6] or in artificial neural networks [7] to be an appropriate function to introduce the needed non-linearity (as seen in Figure 3: a stronger decay or a stronger ascent even for egocentric edge networks with moderate differences in the number of one hop and two hop neighbors) that is not incorporated in the WNH formulation. Also, a sigmoid formulation will transform the input (WNH values) to an output that is confined in the range of (0,...,1). Figure 3 also presents the computation of the OTH value for edge (2, 3) in the example graph of Figure 1.

$$WNH(u) = |N(u)| - 2 * |Ego_N(u,v) - N(u)| \qquad \text{..... (2)}$$
$$WNH(v) = |N(v)| - 2 * |Ego_N(u,v) - N(v)|$$
$$OTH(u,v) = \frac{1}{1 + e^{-(WNH(u)+WNH(v))}} \qquad \text{......... (3)}$$

2.3. Neighborhood Overlap (NOVER)

The Neighborhood Overlap (NOVER [87]) score (see formulation 4 below) for an edge (*u*, *v*) is simply the ratio of the number of common neighbors for the two end vertices *u* and *v* and the total number of neighbors of the two end vertices *u* and *v* with each neighbor counted only once (excluding *u* and *v* themselves). Figure 4 presents the computation of the NOVER score for edge (2, 3) in the example graph of Figure 1.

$$NOVER(u,v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)| - 2} \qquad \text{............ (4)}$$

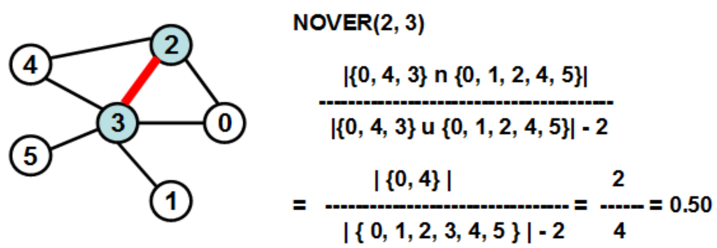


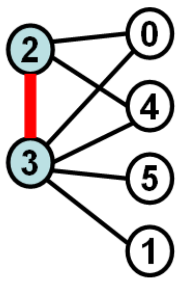
Figure 4. Example to Compute the Neighborhood Overlap (NOVER) for an Edge

2.4. Complement of Bipartivity Index (BPI')

Bipartivity Index [6] is a quantitative measure of the extent with which we could the group the vertices of a graph into two disjoint partitions such that there are as minimal edges, if any, between vertices within the same partition. Edges between vertices within the same partition are called frustrated edges [6]. For any network graph, the BPI values range from 0.5 to 1 [6]. A graph is truly bipartite if there are no frustrated edges; the BPI for such a graph is 1 [6]. Graphs with higher BPI (closer to 1) could be referred to as being *close-to-bipartite* [6]. The BPI for a graph is computed [6] using the following formulation (5), where λ_1, λ_2, λ_3, ..., λ_n are the eigenvalues of the adjacency matrix of the graph of '*n*' vertices.

$$BPI(G) = \frac{\sum_{j=1}^{n} \cosh(\lambda_j)}{\sum_{j=1}^{n} \cosh(\lambda_j) + \sum_{j=1}^{n} \sinh(\lambda_j)} \quad \text{............ (5)}$$

In [5], we proposed that the egocentric network of an edge (*u*, *v*) could be modeled as a close-to-bipartite graph with just one frustrated edge: the end vertices *u* and *v* form one partition and their neighbor vertices form the other partition. The only frustrated edge in such a close-to-bipartite graph would be the edge (*u*, *v*) itself; hence, such graphs tend to have a larger BPI. Nevertheless, we observed the following properties [5] for egocentric edge networks modeled as close-to-bipartite graphs: For two egocentric networks with the same number of edges, the larger the number of shared neighbors, the lower the BPI (i.e., larger BPI’ = 1 – BPI) and vice-versa. Also, For two egocentric networks with the same number of shared neighbors, the fewer the number of unshared neighbors, the lower the BPI (i.e., larger BPI’ = 1 – BPI) and vice-versa. Both these observations lead us to propose BPI' of the egocentric network of an edge as a measure of the extent of shared neighborhood, which is in turn a measure of link stability. Figure 5 presents the computation of the BPI' value for the egocentric network of edge (2, 3) in the example graph of Figure 1.



Eigenvalues	cosh	sinh
-1.999	3.762	-3.627
-1.343	2.045	-1.784
0.0	1.0	0.0
0.0	1.0	0.0
0.529	1.143	0.554
2.813	8.365	8.305
Sum	17.315	3.448

BPI (2, 3)

17.315

{17.315 + 3.448}

BPI’ (2, 3)
= 1 – BPI (2, 3)
= 1 – 0.834 = 0.166

Figure 5. Example to Illustrate the Computation of the BPI' Score for an Edge

2.5. Comparison of the Link Stability Scores

In this sub section, we present the link stability scores obtained for the edges in the example graph of Figure 1 with respect to the four metrics BPI', NOVER, NND and OTH (see Figure 6-a) as well as analyze the correlation between any two of the four metrics (see Figure 6-b). We observe relatively larger values for the rank-based correlation coefficients (compared to the Pearson's prediction correlation coefficients) for the NOVER and OTH measures vs. BPI' as well as between NOVER and OTH themselves. On the other hand, the Pearson's prediction correlation coefficients [9] for the NND measure vs. the BPI', NOVER and OTH measures are larger than that of the Spearman's rank-based correlation coefficients [9]. Nevertheless, both Spearman's and Pearson's correlation coefficients are individually above 0.80 for any two of the three combinations of the LSS measures BPI', NOVER and OTH. Though NND has a weaker correlation with BPI', we expect the prediction accuracy of BPI' to improve with the inclusion of NND in the regression formulation along with NOVER and OTH. Note that the example graph of Figure 1 is a toy graph. The extent of influence (on a relative basis) of the computationally-light metrics on BPI' could be different for the simulations conducted with mobile sensor networks.

Edge	BPI'	NOVER	OTH	NND
(0, 1)	0.1112	0.3333	0.0179	0.8165
(0, 2)	0.1112	0.3333	0.0179	0.8165
(0, 3)	0.1660	0.5000	0.1192	0.7906
(1, 3)	0.1660	0.5000	0.1192	0.7906
(1, 5)	0.0889	0.2000	0.0025	0.5656
(2, 3)	0.1660	0.5000	0.1192	0.7906
(2, 4)	0.0889	0.2000	0.0025	0.5656
(3, 4)	0.1455	0.3333	0.0179	0.5773
(3, 5)	0.1455	0.3333	0.0179	0.5773
(4, 5)	0.2022	0.6000	0.5000	0.7483
(4, 6)	0.1660	0.5000	0.1192	0.7906
(4, 7)	0.1660	0.5000	0.1192	0.7906
(5, 6)	0.1660	0.5000	0.1192	0.7906
(5, 7)	0.1660	0.5000	0.1192	0.7906
(6, 7)	0.2054	1.0000	0.5000	1.4142

(a) Link Stability Scores of the Edges

Spearman's Rank Correlation

	BPI'	NOVER	OTH	NND
BPI'		0.9919	0.9909	0.4092
NOVER			0.9989	0.4929
OTH				0.4733
NND				

Pearson's Prediction Correlation

	BPI'	NOVER	OTH	NND
BPI'		0.8697	0.9348	0.5847
NOVER			0.8038	0.8984
OTH				0.5917
NND				

(b) Correlation between any Two Metrics

Figure 6. Comparison of the Link Stability Scores for the Example Graph of Figure 1

3. PROCEDURE FOR MULTI-VARIABLE REGRESSION

In this section, we describe the procedure to develop an in-situ multi-variable regression model wherein we build the model and test it within a simulation run itself. Our approach comprises of dividing the time period of the simulation run into two phases: a training phase followed by the testing phase.

In the first phase (called the training phase), we build a dataset comprising of the NOVER, OTH and NND metric scores for the edges and their corresponding actual BPI' values using the egocentric edge network approach and the algorithms explained in Section 2. We sample the network for every one second, for a total of 100 seconds. For each of these sampling time instants, we gather data for the above four metrics for each of the edges in the network. We run the multi-variable linear regression algorithm (available as part of the Apache Java package: http://commons.apache.org/proper/commons-math/javadocs/api-3.6.1/index.html) and estimate the parameters of the regression model. We build regression models involving {NOVER} vs. BPI', {NOVER, OTH} vs. BPI' and {NOVER, OTH, NND} vs. BPI'. For all the scenarios, we observe the {NOVER, OTH, NND} vs. BPI' linear regression model to incur a significantly larger R^2 value than the models involving a subset of these three computationally-light metrics.

In the second phase (called the testing phase), we predict the BPI' for an egocentric edge network at a particular time instant using the coefficients of the regression model for the NOVER, OTH and NND metrics computed for that time instant. Using the predicted BPI' values as edge weights, we determine a maximum spanning tree-based data gathering tree and use the tree for data gathering as long as it exists. When the DG tree breaks due to the failure of one or more links at a particular time instant, we predict the BPI' value for the edges at that time instant using the NOVER, OTH and NND values for that time instant and the regression coefficients. We continue this process for the rest of the simulation time period (for 400 seconds).

Finally, for comparison purposes, we determine a sequence of DG trees using the actual BPI' values of the edges and compare the lifetimes of these DG trees with those determined using the predicted BPI' values. We also determine the performance run time of the procedures to determine the DG trees using the actual and predicted BPI' values. The regression-based approach involves an overhead to build the dataset during the first 100 seconds of the simulation time (which is totally 500 seconds in this research); but, we expect this time overhead to get compensated for the remaining 400 seconds of the simulation during which we just determine the computationally-light NOVER, OTH and NND metrics and predict the value for BPI' using the coefficients for these metrics in the multi-variable regression model.

4. SIMULATIONS

We conducted exhaustive simulations to evaluate the effectiveness of the multi-variable linear regression model to predict the BPI' values using the computationally-light NOVER, OTH and NND metrics as well as to compare the performance of the DG trees determined based on the actual vs. predicted BPI' values with respect to DG tree lifetime and algorithm run time. The operating conditions used are as follows: network density (50 nodes: low-moderate density; 100 nodes: moderate-high density) and node mobility (the maximum velocity for any node in the network v_{max} is varied with values of 1 m/s, 5 m/s, 10 m/s and 30 m/s, representing mobility levels of low, low-moderate, moderate-high and high respectively). We used the Random Waypoint model [10] as the mobility model. For each combination of network density and node mobility, we generated 10 mobility profile files for a simulation time period of 500 seconds.

For the training/first phase of the regression-based approach, we sampled the network for every one second and determined the NOVER, OTH, NND and BPI' values of the egocentric edge networks for each of these time instants. We build a huge dataset comprising the NOVER, OTH and NND metric as the independent variables and the BPI' metric as the dependent variable. The three-variable linear regression model built during this phase is used in the testing/second phase to predict the BPI' values for a simulation time period of 500 seconds. To get an idea of the magnitude of the regression coefficients and analyze their positive/negative impact on the value for the BPI' metric, we average the values of the regression coefficients over the period of the training time and all the mobility profile files for a particular operating condition. The simulations were conducted on a Dell Precision M4600 laptop with Intel Core i7-2620M CPU @ 2.7 GHz with an installed memory (RAM) of 8 GB.

The results from the simulation studies are as follows: The regression model (based on the average values for the regression coefficients) built for low-moderate density networks (of 50 nodes) is: BPI' = 0.1482 + 0.2453 * NOVER + 0.0495 * OTH - 0.1901 * NND and for moderate-high density networks (of 100 nodes) is: BPI' = 0.2229 + 0.2086 * NOVER + 0.0397 * OTH - 0.2878 * NND. The R^2 values for the regression models are 0.92 and 0.87 respectively for the low-moderate and moderate-high density networks. Note that the R^2 values of the regression models with just one or two of the three computationally-light metrics as independent variables were less than 0.75. From the above equations, we could infer the NOVER metric to have a relatively larger positive influence on the value for the BPI' metric and the NND metric to be the only metric to have negative influence on BPI'. The OTH metric has the relatively lowest influence (slightly positive influence) among the three computationally-light metrics. Table 1 shows the results for the average DG tree lifetime and run time (both measured in seconds, physical time) of the procedures to determine the DG trees based on the actual vs. predicted BPI' values. For each operating condition, the cells highlighted correspond to the larger DG tree lifetime or lower algorithm run time (either of which are preferred outcomes) with respect to the actual vs. predicted BPI' metric values.

Table 1. Simulation Results

	Low-Moderate Density Networks (50 nodes)			
v_{max}	DG Tree Lifetime		Algorithm Runtime	
	BPI'_{actual}	BPI'_{pred}	BPI'_{actual}	BPI'_{pred}
1 m/s	8.13 sec	7.10 sec	0.92 sec	3.31 sec
5 m/s	2.31 sec	2.11 sec	2.98 sec	4.19 sec
10 m/s	1.29 sec	1.14 sec	4.66 sec	4.48 sec
30 m/s	0.56 sec	0.52 sec	10.89 sec	6.74 sec
	Moderate-High Density Networks (100 nodes)			
v_{max}	DG Tree Lifetime		Algorithm Runtime	
	BPI'_{actual}	BPI'_{pred}	BPI'_{actual}	BPI'_{pred}
1 m/s	13.58 sec	7.55 sec	7.12 sec	49.96 sec
5 m/s	2.97 sec	1.96 sec	33.61 sec	49.92 sec
10 m/s	1.62 sec	1.09 sec	63.03 sec	59.13 sec
30 m/s	0.66 sec	0.50 sec	165.17 sec	80.85 sec

As the regression-based approach involves an overhead of computing the BPI' metric values for the first 100 seconds (100 sampling time instants), this time overhead actually dominates the algorithm run time when the network topology does not change much (for v_{max} values of 1 m/s and 5 m/s). On the other hand, as node mobility increases, the actual BPI' values have to be frequently determined, and the regression-based approach of just predicting the BPI' values (based on the regression coefficients estimated in the first 100 seconds) and determining the DG trees (any time after the first 100 seconds) starts incurring a relatively lower run time. For networks of moderate-high density, the algorithm run time with respect to BPI'_{actual} increases significantly with increase in v_{max}. The lifetimes of the BPI'_{pred} -based DG trees are 87-92% and 55-75% of the BPI'_{actual} -based DG trees for low-moderate and moderate-high density networks respectively. For both scenarios of network density, the difference in the lifetime between the BPI'_{actual} vs. BPI'_{pred} DG trees decrease with increase in node mobility. With respect to algorithm run time, the BPI'_{actual} -based approach could take as low 14% of the algorithm run time for the BPI'_{pred} -based approach in scenarios of low node mobility and moderate-high network density, and as high as 204% of the algorithm run time for the BPI'_{pred} -based approach in scenarios of high node mobility and moderate-high network density. We could thus conclude to use the BPI'_{pred} -based approach for scenarios of very high node mobility and networks of any density, and employ the BPI'_{actual} -based approach in networks of low-moderate node mobility and networks of any density.

5. CONCLUSIONS

The high-level contribution of this paper is the idea of using two or more computationally-light metrics to predict the value for a computationally-heavy metric (using multi-variable linear regression) to quantify link stability in mobile sensor networks. In this pursuit, we propose two computationally-light metrics (Normalized Neighbor Degree, NND and One Hop Two Hop Neighbors, OTH) that are used along with Neighborhood Overlap (NOVER) to develop a multi-variable regression model to predict the values for the computationally-heavy BPI' (complement of Bipartivity Index) metric. The regression coefficients are observed to appreciably vary with network density (for fixed node mobility) and are independent of node mobility (for fixed network density). Overall, we could thus come up with two regression models, one model for networks of low-moderate density and another model for networks of moderate-high density.

The magnitude and sign of the regression coefficients for the three computationally-light metrics in these models indicate the NOVER metric to have a stronger positive influence on the BPI' values and the NND metric to have a moderately negative influence on the BPI' values; the OTH metric has the least significance on the BPI' values (imparts only a slightly positive influence). The R-square values of the regression models are above 0.85 for all scenarios of network density and node mobility. We build the regression models in-situ (within a simulation run) and there is no need for external/additional datasets. Performance wise, we observe the regression-based approach of predicting the BPI' values to be time efficient in scenarios of moderate-high node mobility, irrespective of network density. With regards to the accuracy of the prediction, we observe the lifetimes of the BPI'_{pred} -based data gathering (DG) trees to be 87-92% and 55-75% of the BPI'_{actual} -based DG trees for low-moderate and moderate-high density networks respectively.

REFERENCES

[1] N. Meghanathan, "A Comprehensive Review and Performance Analysis of Data Gathering Algorithms for Wireless Sensor Networks," International Journal of Interdisciplinary Telecommunications and Networking (IJITN), vol. 4, no. 2, pp. 1-29, April-June 2012.

[2] Abolhasan, M., Wysocki, T., Dutkiewicz, E., "A Review of Routing Protocols for Mobile Ad hoc Networks," *Ad hoc Networks*, vol. 2, no. 1, pp. 1-22, 2004.

[3] W. Su and M. Gerla, “IPv6 Flow Handoff in Ad hoc Wireless Networks using Mobility Prediction,” *Proceedings of the IEEE Global Telecommunications Conference*, pp. 271-275, December 1999.

[4] N. Meghanathan, "Link Expiration Time and Minimum Distance Spanning Trees based Distributed Data Gathering Algorithms for Wireless Mobile Sensor Networks," *International Journal of Communication Networks and Information Security*, vol. 4, no. 3, pp. 196-206, December 2012.

[5] N. Meghanathan, "Complex Network Analysis-based Graph Theoretic Metrics to Determine Stable Data Gathering Trees for Mobile Sensor Networks," *The Computer Journal*, vol. 61, no. 2, pp. 199-222, February 2018.

[6] E. Estrada and J. A. Rodriguez-Velazquez, "Spectral Measures of Bipartivity in Complex Networks," *Physical Review E* 72, 046105, pp. 1-6, 2005.

[7] D. Easley and J. Kleinberg, *Networks, Crowds and Markets: Reasoning about a Highly Connected World*, 1st Edition, Cambridge University Press, Cambridge, UK, 2010.

[8] N. Meghanathan, "Neighborhood Overlap-based Stable Data Gathering Trees for Mobile Sensor Networks," *International Journal of Wireless Networks and Broadband Technologies*, vol. 5, no. 1, pp. 1-23, January-June 2016.

[9] G. Strang, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, 5th Edition, June 2016.

[10] C. Bettstetter, H. Hartenstein and X. Perez-Costa, “Stochastic Properties of the Random-Way Point Mobility Model,” *Wireless Networks*, vol. 10, no. 5, pp. 555-567, September 2004.