Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-2 (2022) 408–413

# The value of information for dynamic decentralised criticality computation ⋆

**Yaniv Proselkov* Manuel Herrera****
**Marco Perez Hernandez*** Ajith Kumar Parlikad******
**Alexandra Brintrup†**

*Institute for Manufacturing, Dept. of Engineering, University of Cambridge*
*yp289@cam.ac.uk; **amh226@cam.ac.uk; ***mep53@cam.ac.uk; ****aknp2@cam.ac.uk; †ab702@cam.ac.uk

**Abstract:** Smart manufacturing uses advanced data-driven solutions to improve performance and operations resilience requiring large amounts of data delivered quickly, enabled by telecom networks and network elements such as routers or switches. Disruptions can render a network inoperable; avoiding them requires advanced responsiveness to network usage, achievable by embedding autonomy into the network, providing fast and scalable algorithms that use key metrics to manage disruptions, such as impact of failure in a network element on system functions. Centralised approaches are insufficient for this as they need time to transmit data to the controller, by which time it may have become irrelevant. Decentralised and information bounded measures solve this by placing computational agents near the data source. We propose an agent-based model to assess the value of the information for calculating decentralised criticality metrics, assigning a data collection agent to each network element, computing relevant indicators of the impact of failure in a decentralised way. This is evaluated by simulating discrete information exchange with concurrent data analysis, comparing measure accuracy to a benchmark, and with measure computation time as a proxy for computation complexity. Results show losses in accuracy are offset by faster computations with fewer network dependencies.



*Keywords:* Computational Science; Discrete-event Simulation; Dynamic Systems; Intelligent Diagnostic Methodologies; Large Scale Multi-agent Systems; Multi-agent Simulation; Visibility; Criticality

## 1. INTRODUCTION

Manufacturing processes have become more data-driven and dependent on interconnection of multiple facilities for efficient decision-making, thus telecom infrastructure is as pervasive a component of manufacturing industries as the powergrid and other critical infrastructures. Telecom infrastructures are physical networks that support internet, telephony, and other digital services by facilitating data transfers between users. Infrastructures are represented by graphs, with network elements, such as routers or switches, as nodes and connections as edges. In such flow networks, data packet congestion at a node may cause node failure. How to monitor the impact of disruptions such as congestion on the network - criticality - is important for network behaviour control, which must be accurate and fast in networks that are functioning near capacity, as expected for future backbone networks (Moura and Hutchison (2019)).

In centralised network monitoring and criticality computation, the central computational resources need information on the whole network, creating a *criticality measure* (CM) (Salazar et al. (2016); Fang and Zio (2013)), requiring live and dynamic node topology and attribute data to respond to behavioural shifts. The increased data cause longer computational times, so conclusions arrived at a given point in time lose relevance, and promote critical events if too much data is transferred. The amount of data used for a CM should thus be minimised while preserving meaning, achievable by imposing a limited bound around a given node, and computing criticality in a decentralised manner. This shortens data paths and reduces complexity by taking information from a small region around a given node. We call these *information bounded* CMs (IB-CMs). This IB-CM can be approximated with a *centrality measure* used as a *criticality estimate* (CE), as both define importance within a multicomponent system (Birnbaum (1968)). The more accurate and more efficiently computed an IB-CM is, the more valuable the information used in its calculation is, with the *value of information* is a multivariate measure composed of the accuracy and efficiency of a given IB-CM under some information provision.

This paper builds on Proselkov et al. (2020) to outline a method to assess the accuracy and computational efficiency of IB-CMs that change with time, with respect to a novel benchmark estimate of dynamic criticality under

⋆ This research was supported by the EPSRC and BT Prosperity Partnership project: Next Generation Converged Digital Infrastructure, grant number EP/R004935/1, and the UK Engineering and Physical Sciences Research Council (EPSRC) Doctoral Training Partnership Award for the University of Cambridge, grant number EP/R513180/1.


10.1016/j.ifacol.2022.04.228

different *communication paradigms* (CPs). These IB-CMs are designed for homogeneous flow networks. A prototype is presented that uses classic centrality measures as a stand in for CMs and IB-CMs, with real network topology on the use case of a telecom simulation model.

## 2. LITERATURE REVIEW

Network topology affects routing and resilience to disruption since shorter distances give quicker transfers. Criticality, defined as the impact of a node's inactivity on the operation of a network, evaluated by network connectivity in telecoms, is a key factor in understanding network resilience (Lü et al. (2016); Herrera et al. (2020)), and can be estimated using the current network state (Proselkov et al. (2020)). Criticality can inform prioritisation in network prognostics for proactive maintenance. Many criticality measures are based on centrality measures, including betweenness (Freeman (1977)); eigencentrality; and degree centrality. The first two are centralised, needing each node to take information from all nodes. Degree centrality needs each node to know the number of their neighbours.

Efficient decentralised computation approaches for understanding network criticality are important for networks operating under stress, (Cetinkaya and Sterbenz (2013)). Cascade failure may also occur within regularly functional systems due to random errors, as in January 1990, where 114 switching nodes of the AT&T network successively went down due to a wrong reset signal (Neumann (1995)).

Nodes within telecoms networks provide information of their state either by transmitting to a supervisory node, which facilitates centralised centrality calculation, or with each other, which facilitates distributed centrality calculation. They can achieve decentralised communication through broadcasting to all neighbours their node ID, the value, and topological information including the travel history of the data packet, and previously broadcasted packets that are known to remain in motion, (Lehmann and Kaufmann (2003)). This takes at least the minimum distance between two nodes to be completed in reality.

Experimental evidence suggests increased computational efficiency and satisfactory performance of information bounded network measures as in Ercsey-Ravasz and Toroczkai (2010). This details the relationship of the depth of the information bound and size of the value distribution of the associated bounded betweenness measures. The value distribution increases exponentially with the depth of the bound up to mean geodesic length before decreasing, suggesting meaningful sensitivity at the mean geodesic length. Tests were conducted on scale-free and random graphs. These only have one cluster, so it is expected that the ideal depth may be the mean cluster geodesic length.

Other papers give examples of limited range criticality and centrality for static measures (Wehmuth and Ziviani (2011); Chen et al. (2012); Nanda and Kotz (2008); Kermarrec et al. (2011); Dinh et al. (2010); Proselkov et al. (2020)) and for dynamic distributed criticality measures. All show accuracy despite limited boundaries. However, no large scale analysis on the relative efficiency via computation time and accuracy has not yet been conducted for dynamic criticality measures.

## 3. METHODS

### 3.1 Telecom simulation model

The network topology is generated, creating the graph, $G = (V, E)$ where $V$ is nodes and $E$ is edges. The discrete information packet exchange simulator is then run [1] with short range dependence, meaning random nodes generate data packets independently according to a Poisson distribution with random destinations (Veres and Boda (2005)). As this is an information flow network, a timestep is how long it takes for information to traverse one edge. Packets traverse the network, stored and routed along nodes on the way to their destination, where they are removed from the system. Nodes and edges each have a fixed capacity which gets filled up over time since it takes time to process packets at nodes and transmit them between nodes, with processing and transmission time as fixed model input parameters. This process is terminated after either a fixed number of timesteps or until the network is too congested to function. The nodes each have a backlog capacity of $\phi$. The simulation produces a time series over $T$ of each node's queued up data packet backlog, where the size of the queue held by node $u \in V$ at time $t \in T$ is $\phi_u^t$.

We then examine how nodes would behave if they were receiving and processing network data in real time in an agent based simulation [2] , an independent agent situated at each node. Depending on our monitoring data CP, which determines how up-to-date information is (*currentness*), different nodes get different information regarding others depending on their position in the network. We investigate three CPs, named *instant*, *constant*, and *periodic*.

For a pair of nodes, $u, v \in V$, there is a path $p_{uv} \subseteq G$ from $u$ to $v$ if, for some $n \in \mathbb{Z}^+$, there exists an ordered sequence of nodes, $(u, (u_i)_{i=0}^n, v) \subseteq V$ with edges $\{uu_0, (u_i u_{i+1})_{i=0}^{n-1}, u_n v\} \subseteq E$, or $uv \in E$. If there is a path from $u$ to $v$, $u$ gets information about $\phi_v^t$. The queue at time $t$ that the agent at $u$ believes $v$ has is the *perceived queue*, $q_{uv}^t \in Q$. If $u = v$ this is $q_u^t$. Centrality calculation takes time so the time from transmission to output is always greater than transmission to receipt between nodes, thus nodes must compute centralities at a lower frequency than the CP dictates to avoid losing currentness. This period between calculations is $\mu$, the *monitor interval*.

*Instant communication* is a simplification, assuming monitoring data is transferred instantly, where for all $u, v \in V$, $q_{uv}^t = q_v^t$. This is a base case, only achievable if monitoring data transfer became so fast as to be insignificant.

*Constant communication* has nodes declare their queues every timestep, This data traverses the network normally, because this declaration is a low bandwidth operation. For shortest path (geodesic) of length $n$ between $u$ and $v$, information from $v$ takes $n$ timesteps to reach $u$, so $u$ perceives $v$ $n$ timesteps late, so $q_{uv}^t = q_v^{t-n}$, so distant nodes give less accurate, valuable or relevant information.

*Periodic communication* has nodes declare their queues and perceived queues at the same frequency as they calculate their centrality. This corresponds to some aggregation

[1] using a Python package called "Anx" (Likic and Shafi (2018)),
[2] using a Python package called "Mesa", (Kazil et al. (2020))

of the functions within higher order control functions that use centralities as inputs. For monitor interval $\mu$ and $u$ and $v$ with length $n$ geodesic, $q_{uv}^t = q_v^{t-\mu n}$, as each queue pass is $\mu$ timesteps.

With each CP, nodes receive perceived queues of others in the network. These values are used to inform dynamic, queue dependant CEs, which are calculated with the adjacency matrix, and so with respect to edge weight rather than node weight, which we assign according to the following steps. First the graph is redefined as directed, such that for $(uv), (vu) \in E$, $(uv) \neq (vu)$. For a node $u$, for all $v \in \Gamma_1(u)$, the weight of edge $(uv)$ is

$$(uv)_q^t = q_u^t / |\Gamma_1(u)|, \tag{1}$$

because larger neighbourhoods give nodes more chances to emit data packets and distribute load among them, and it accounts for the respective queues of node pairs since for all $(uv)$ there exists a $(vu)$ produced under the same rules.

The following subsection describes the data analysis carried out with the data provided according to each CP.

### 3.2 Centrality Measure CEs

According to each of the above CPs, the data delivered to each agent situated at a node is used to compute centrality measures over time as proxies and estimates of criticality. In this initial study standard centrality measures are used, with weighted and bounded extensions. The unweighted measures only take topological data, whereas weighted measures adjust their outputs according to the perceived queues for each node. Unbounded, or *sociocentric*, measures take information from the whole network and stand in for CMs, while bounded, or *egocentric* measures take information from a limited region around a given node and stand in for IB-CMs.

We define the information boundary around a node by the geodesic distance. For a node, $u \in V$, the set of nodes $i$ edges away is $\Gamma_i(u) \subset V$, where $\Gamma_1(u)$ is the neighbourhood of $u$. The set of nodes at most $i$ edges away from $u$ is $H_i(u) = \bigcup_{j=1}^{i} \Gamma_j(u)$. If $u$ has an information boundary at distance $i$, it takes information from $H_i(u)$.

*Degree Centrality:* *Unweighted Degree Centrality* for a node $u$ counts the number of neighbours. It is defined as $C_{\mathrm{d}}^{\mathrm{u}}(u)_t = C_{\mathrm{d}}^{\mathrm{u}}(u) = |\Gamma_1(u)|$.

*Weighted Degree Centrality* counts each node as many times as their perceived queue lengths. It is dynamic and defined as $C_{\mathrm{d}}^{\mathrm{w}}(u)_t = \sum_{v \in \Gamma_1(u)} q_{uv}^t$.

*Betweenness Centrality:* All distinct paths with the same length and the minimum number of elements are geodesics. The number of geodesics from $v$ to $w$ is $\rho_{v,w} : V \to \mathbb{Z}^+$, and the number of geodesics from $v$ to $w$ passing through $u$ is $\rho_{v,w|u} : V \to \mathbb{Z}^+$.

*Unweighted Sociocentric Betweenness Centrality* (Freeman (1977)), tracks pathway disruption potential. It is static, calculating the fraction of shortest paths between all node pairs passing through the subject node.

*Unweighted Egocentric Betweenness Centrality* measures the betweenness of a bounded region surrounding a node. It correlates strongly with sociocentric betweenness (Marsden (2002)), and is computable in a decentralised manner. For node $u \in V$ it measures the betweenness of the induced subgraph of $H_i(u)$, such that

$$C_{\mathrm{b}}^{\mathrm{ue}}(u)_t = C_{\mathrm{b}}^{\mathrm{ue}}(u) = \sum\nolimits_{v,w \in H_i(u)} \rho_{v,w|u} / \rho_{v,w}.$$

*Weighted Sociocentric Betweenness Centrality* uses a weighted shortest path parameter. $P_{vw}$ is the set of shortest paths between nodes $v$ and $w$, and using Eqn. (1) the CE is defined as

$$\begin{aligned} \omega_{v,w}^t &= \sum\nolimits_{p_{vw} \in P_{vw}} \sum\nolimits_{(st) \in p_{vw}, (st) \in E} (st)_q^t; \\ \omega_{v,w|u}^t &= \sum\nolimits_{u \in p_{vw} \in P_{v,w}} \sum\nolimits_{(st) \in p_{vw}, (st) \in E} (st)_q^t, \end{aligned} \tag{2}$$

giving weighted sociocentric betweenness centrality as

$$C_{\mathrm{b}}^{\mathrm{ws}}(u)_t = \sum\nolimits_{v,w \in V, u \neq v \neq w} \omega_{v,w|u}^t / \omega_{v,w}^t. \tag{3}$$

*Weighted Egocentric Betweenness Centrality* takes Eqn. (3) but over $H_i(u)$.

*Eigencentrality:* *Unweighted Sociocentric Eigencentrality* captures the connectivity of the network, valuing nodes with more connections to well connected nodes. The number of edges between nodes $u_i$ and $u_j$ is $a_{i,j}$, displayable in a matrix, $A_G \in M_n(\{0,1\})$, the *adjacency matrix*, where

$$A_G = (a_{i,j}) = \begin{cases} 1, & (i,j) \in E \\ 0, & i = j, \end{cases}$$

for the matrix, $G$. The eigencentralities of the nodes in the network are found for the largest eigenvalue, $\lambda_G$, with

$$A_G \boldsymbol{x} = \lambda_G \boldsymbol{x}, \tag{4}$$

and the CE is the solution to Eqn. (4), numerically solved via power iteration, or Von Mises iteration (von Mises and Pollaczek-Geiringer (1929)).

*Unweighted Egocentric Eigencentrality* is the solution to Eqn. (4) over $H_i(u)$ rather than over $G$.

*Weighted Sociocentric Eigencentrality* uses the directed network with edge weights as defined by Eqn. (1). The adjacency matrix becomes dynamic and temporally dependant, such that for $A_G^t \in M_n(\mathbb{Z}^+)$,

$$A_G^t = (a_{i,j}) = \begin{cases} (u_i u_j)_q^t, & (i,j) \in E \\ 0, & i = j. \end{cases} \tag{5}$$

$A_G$ in Eqn. (4) is then replaced by $A_G^t$ in Eqn. (5).

*Weighted Egocentric Eigencentrality* uses $A_G^t$ from Eqn. (5). For node $u_j$ it is over $H_i(u_j)$, not $G$, creating

$$C_{\mathrm{e}}^{\mathrm{we}}(u_j)_t = (A_{H_i(u)}^t \boldsymbol{x})_j = (\lambda_{H_i(u)}^t \boldsymbol{x})_j.$$

These CEs will be used as proxies for criticality measures. To compute the value of information as processed through each measure, we now outline a validation method.

### 3.3 Validation Method

The measures above must be validated as correctly approximating dynamic criticality within the network. A validation function must determine at any timestep the similarity of our CE to a benchmark and its period of relevance. Criticality measures the impact of failure, which must be defined, and for how long the effects of some action can be said to have been caused by a previous one. Analysis is conducted post hoc, using data that is neither limited by the imperceptibility of the future nor communication constraints. We take linear functions of the total queue sizes of the whole network, using $\Phi^t = \sum_{u \in V} \phi_u^t$. We also find a time range for which we have sufficient confidence that all network states are sufficiently dependant on eachother.

*Ideal Time Horizon* This is a moving time window, bisected by the present timestep, where the window's start sufficiently influenced all timesteps up to the present, and the present will sufficiently influence all timesteps up to the window's end. With it, we can find how far must a CE look into the future to sufficiently capture both the current network state and its influence. We iterate over a fixed number, $h_{\text{test}}$, of time horizon windows, $h_i$, less than half the simulation length, $t_{\max}$, where $h_i = it_{\max}/(2h_{\text{test}})$, and take moving averages over $\Phi^t$ for each width $h_i$, so

$$\mathrm{MA}^t_{\Phi;h_i} = \begin{cases} \sum_{t-i}^{t} \Phi^t / i & t \geq i; \\ \emptyset & t < i, \end{cases}$$

and $\mathrm{MA}_{\Phi;h_i}$ is the time series made up by $\mathrm{MA}^t_{\Phi;h_i}$. Then for all $t$ such that $\mathrm{MA}^t_{\Phi;h_i}$ exists, we take the absolute difference between $\mathrm{MA}^t_{\Phi;h_i}$ and $\Phi^t$, such that

$$\mathrm{MAD}^t_i = \begin{cases} |\mathrm{MA}^t_{\Phi;h_i} - \Phi^t| & t \geq i; \\ \emptyset & t < i, \end{cases}$$

and get the sum of absolute differences, $\mathrm{SAD}_i = \sum_t \mathrm{MAD}^t_i$. Normalised, this is $\mathrm{NSAD}_i = \mathrm{SAD}_i / \max_{i=1}^{h_{\text{test}}} \mathrm{SAD}_i$. Iterating through $\mathrm{NSAD}_i$ in ascending $i$, we obtain $g_i = h_{\text{test}}(\mathrm{NSAD}_{i+1} - \mathrm{NSAD}_i)$. The ideal time horizon is where the relative gain in error by a wider window is large enough to suggest that all smaller window sizes cover regions with significant influence over eachother. Beyond that, since error gain slows down, one cannot confidently claim events are the direct consequence of the current time. This confidence, the *validation threshold*, is an independent parameter, $c$, with which we define the ideal time horizon, $h$ for the first $i$ where one of the following conditions is fulfilled, where if the last case is reached we must test more windows or increase the confidence threshold:

$$h = \begin{cases} \lfloor i/2 \rfloor & g_i \leq c; \\ \lfloor (i-1)/2 \rfloor & g_i < 0; \\ \emptyset & i = h_{\text{test}}. \end{cases}$$

*Comparison Accuracy Function* We compare CEs to a benchmark measure of criticality, defined as the change in network operation induced by any network state changes. Dependencies are sufficiently large for all timesteps at most $h$ timesteps far from eachother, so impacts occur over a meaningful timescale of $h$. Impact at time $t$ is the change over $h$ timesteps across $t$ scaled by the built up queues at time $t$, since a heavily used system has more to lose than an underused one. We obtain a moving average with window width $h$, $\mathrm{MA}_{\Phi;h}$ and produce a time series of scaled differences across a time horizon, $\mathrm{THD}_t = \mathrm{MA}^t_{\Phi;h}\left(\mathrm{MA}^{t+h}_{\Phi;h} - \mathrm{MA}^{t-h+1}_{\Phi;h}\right)$. This is normalised to $[0,1]$, creating $\mathrm{NTHD}_t = (\mathrm{THD}_t - \min_t \mathrm{THD}_t)/(\max_t \mathrm{THD}_t - \min_t \mathrm{THD}_t)$, the *criticality benchmark*. For a CE, $C(u)_t$, we calculate the network mean, $\overline{C_t} = \sum_u C(u)_t$, and normalise to get $\mathrm{NC}_t$. Let $\mathcal{T} = \{\tau \in T : \tau = k\mu, k \in \mathbb{Z}^+\}$. The error from the benchmark is $\mathrm{Err}_t = \mathrm{NC}_t - \mathrm{NTHD}_t$, and the root mean squared error is $\mathrm{RMSE} = \sqrt{\sum_{t \in \mathcal{T}} \mathrm{Err}_t^2 / |\mathcal{T}|}$. The lowest RMSE gives the most accurate measure since it most closely follows the benchmark criticality.

For any given CE, the value of information, $V$ for a given dataset, $D$, is the multivariate measure of the reciprocal of the RMSE and the reciprocal of the computation time, Comp, so that it grows with reduced error and increased time efficiency, such that $V(D;\mathrm{CE}) = ((RMSE; D, \mathrm{CE})^{-1}, (\mathrm{Comp}; D, \mathrm{CE})^{-1})$.

## 4. RESULTS AND DISCUSSION

We compared simulation results of instant, constant, and periodic CPs for the accuracy of decentralised, dynamic, and information bounded centrality measures for estimating criticality. Three simulations, one for each CP, using the real topology of the UK outer backbone infrastructure network for a UK telecoms service provider (Fig. 1).

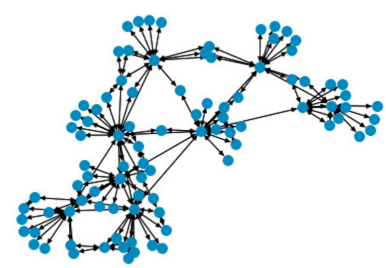


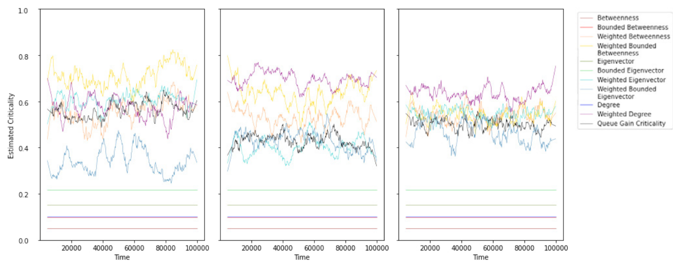
Fig. 1. Outer backbone UK infrastructure network for a large UK service provider

Simulation inputs are in Table 1. We skip 5000 timesteps to avoid degeneracy since we initialise on an empty network.

Table 1. Simulation Inputs

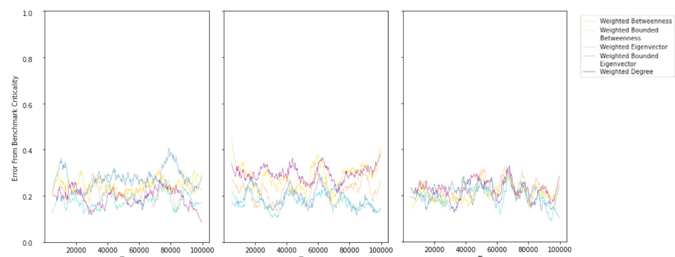
| | |
|---|---|
| Runtime $t_{\max}$ (hundreths of a second) | 100000 timesteps |
| Monitor Interval $\mu$ | 250 timesteps |
| Packet generation rate | 19 packets/timestep |
| Processing delay | 13 timesteps |
| Queue check time | 1 timestep |
| Transmission time | 30 timesteps |
| Node capacity $\phi$ | 128 packets |
| Link capacity | 1024 packets |
| Information visibility boundary | 2 hops |
| Validation threshold $c$ | 0.1 relative difference |
| Window widths tested $h_{\text{test}}$ | 20 windows |
| Ideal time horizon $h$ | 10000 timesteps |

Fig. 2 shows plots of all centralities and the criticality benchmark, $\mathrm{NTHD}_t$, for each CP, filtered using a first order Savitzky-Golay filter. This graph shows substantial difference between outputs for weighted and unweighted measures across all CPs, but further analysis will show similar accuracy. The absolute error, $|\mathrm{Err}_t|$, from $\mathrm{NTHD}_t$ was calculated, the results plotted in Fig. 3. These plots are only for weighted measures since error from static values is a trivial transformation of the criticality benchmark. We can see the relative accuracy of each curve, showing similar accuracy between bounded and unbounded measures. Periodic CP readouts are more closely clustered in terms of accuracy. Error plots are filtered using a first order Savitzky-Golay filter.



(b) Instant (c) Constant (d) Periodic

Fig. 2. CEs and NTHD$_t$s for simulations of each CP.



(b) Instant (c) Constant (d) Periodic

Fig. 3. Error for Weighted CEs and NTHD$_t$s for simulations of each CP.

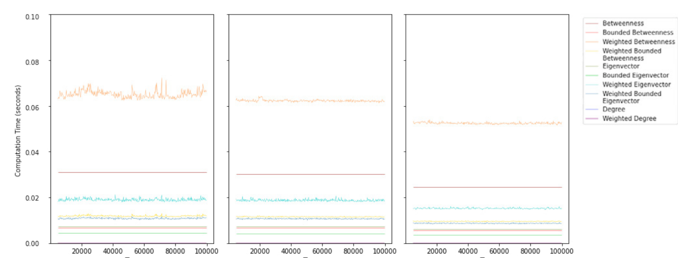
Table 2 shows RMSE from the criticality benchmark for each CE and each CP. Weighted, dynamic CEs largely performed much better than their static counterparts. Boundedness minimally impacted accuracy. Typically, CEs were most accurate for constant CP, followed by periodic then instant CPs. Of the weighted bounded CEs, betweenness was best in periodic CP with large variation between CPs; eigenvector best in constant CPs; and degree in instant CP, and much better than periodic and constant, which show wide difference. This suggests different CEs can be used for different CPs. No weighted bounded CE had more RMSE than 0.35. Betweenness was worst in constant CP with RMSE 0.339, degree best in instant CP with RMSE 0.239. Eigencentrality and betweenness had similar consistency with ranges of 0.086 and 0.085 respectively. All values are similar and low, suggesting combining bounding and dynamicity creates accurate and scalable CEs.

Table 2. Root Mean Squared Error for each CP and CE. Blue is the least error, red is the most.

| | Instant | Constant | Periodic |
|---|---|---|---|
| Betweenness | 0.536 | 0.412 | 0.482 |
| Bnd'd. Betweenness | 0.491 | 0.368 | 0.437 |
| Wtd. Betweenness | 0.263 | 0.27 | 0.28 |
| Wtd. Bnd'd. Betweenness | 0.301 | 0.339 | 0.254 |
| Eigenvector | 0.44 | 0.321 | 0.388 |
| Bnd'd. Eigenvector | 0.381 | 0.268 | 0.332 |
| Wtd. Eigenvector | 0.228 | 0.212 | 0.244 |
| Wtd. Bnd'd. Eigenvector | 0.327 | 0.241 | 0.259 |
| Degree | 0.487 | 0.365 | 0.433 |
| Wtd. Degree | 0.239 | 0.344 | 0.273 |

Boundedness and the time horizon are spacial and temporal efforts to increase relevancy of a given calculation. A sufficiently small information boundary also reduces computational complexity, allowing calculations to take place within the relevant period. In application, the monitor interval should be bounded above by the relevant period, and is typically bounded below by the computation time. This motivates analysing computation time of each measure under each CP. All analyses were completed on Google Colab Pro, a Jupyter notebook service that provides a Python 3 Google Compute Engine backend of an adaptable memory of up to 32GB RAM with 2 virtual CPUs, Intel(R) Xeon(R) @ 2.20GHz.

Fig. 4 shows computation time plots. Limiting information most affects weighted betweenness, which unbounded can take over 0.07 seconds, but bounded may be less than 0.01, close to weighted bounded eigencentrality. Instant and constant computation times are similar for all measures, though instant CP shows variability and intermittent spikes during network congestion, where queues grow due to build-up exceeding processing speed in certain regions. Periodic CP was uniformly faster, which since it places a lighter memory load through lower frequency, may be an artefact of computational stress on the computer during simulation. Dynamicity increases computation time for complex measures, but minimally impacts degree centrality, computed nearly instantly. Means for each measure and CP are shown in Table 3.



(b) Instant (c) Constant (d) Periodic

Fig. 4. Computation time for all CEs for simulations of each CP, measured in seconds.

Table 3. Mean computation time for all CEs for simulations of each CP, measured in seconds. Blue is the fastest, red is the slowest.

| | Instant | Constant | Periodic |
|---|---|---|---|
| Betweenness | 0.03112 | 0.03022 | 0.0245 |
| Bnd'd. Betweenness | 0.00676 | 0.00669 | 0.00554 |
| Wtd. Betweenness | 0.06512 | 0.06231 | 0.05248 |
| Wtd. Bnd'd. Betweenness | 0.01188 | 0.01154 | 0.00951 |
| Eigenvector | 0.00738 | 0.00726 | 0.00612 |
| Bnd'd. Eigenvector | 0.00445 | 0.0043 | 0.00357 |
| Wtd. Eigenvector | 0.01902 | 0.01874 | 0.01523 |
| Wtd. Bnd'd. Eigenvector | 0.01084 | 0.01068 | 0.00869 |
| Degree | 1.30E-05 | 1.09E-05 | 5.19E-06 |
| Wtd. Degree | 1.83E-05 | 1.38E-05 | 6.88E-06 |

## 5. CONCLUSION

In this paper, we reviewed network criticality, communication paradigms, and existing IB-CMs. We introduced a model of simulated decentralised online network measurement under three CPs. We then defined the IB-CMs we use in our analysis (actually CEs). The validation method that determines the error and from there, value of the information was then outlined. Simulations for the various CEs under different CPs and their results were then detailed, showing the viability of information bounded and dynamic criticality estimation. Together this research provides a framework to develop more advanced CEs. Bounding information visibility is a viable method for scalable measures that preserves accuracy while speeding up calculation.

Each CE was shown to have output occupying the same approximate region between CPs, so behaving similarly. This holds true for error curves too, as in Figs. 2 and 3. In fact, error shows to be more constant for the periodic CP, suggesting an advantage in terms of control for that CP.

Computation time of was found to have similar ordering among CEs between CPs, decreasing in variance and magnitude from instant to constant to periodic communication. This may be an artefact of simulating decentralised agents with a central computer. Alternatively, this may support the argument for decentralised computation, since less information frequency improves computation speed in singular agents. Adding interpolation or statistical data generation may then give fine detailed, accurate measures with low data packet load and high responsiveness.

This study simulated normally operating networks. Further research will simulate critical node failure. We expect this will introduce variability in computation times, which may affect monitoring interval selection. Future work will use also use the advanced CEs developed in (Proselkov et al. (2020)), as well as produce case specific, data derived CEs for maximum relevancy. Using these for network control functions will then allow us to learn the value of information function by removing measure dependency. This will be achieved through the validation method defined in this paper measured against a performance metric. We predict this will give useful findings in studies of network homophily, and tools for policy makers when constructing or designing networks with communication.

Beyond the telecom case, this analytic framework will be applicable to other systems with dynamic flow and independent cognitive agents, such as business networks, mail networks, river networks, and more, each a critical support network for any manufacturing system.

## REFERENCES


Birnbaum, Z.W. (1968). On the Importance of Different Components in a Multicomponent System. Technical report, Washington University Seattle Lab of Statistical Research, Seattle.

Cetinkaya, E.K. and Sterbenz, J.P.G. (2013). A Taxonomy of Network Challenges. In *Design of Reliable Communication Networks*. IEEE.

Chen, D., Lü, L., Shang, M.S., Zhang, Y.C., and Zhou, T. (2012). Identifying influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 391(4), 1777–1787.

Dinh, T.N., Xuan, Y., Thai, M.T., Park, E.K., and Znati, T. (2010). On Approximation of New Optimization Methods for Assessing Network Vulnerability. In *IEEE INFOCOM 2010 - IEEE Conference on Computer Communications*, 1–9. IEEE.

Ercsey-Ravasz, M. and Toroczkai, Z. (2010). Centrality scaling in large networks. *Physical Review Letters*, 105(3).

Fang, Y. and Zio, E. (2013). Hierarchical Modeling by Recursive Unsupervised Spectral Clustering and Network Extended Importance Measures to Analyze the Reliability Characteristics of Complex Network Systems. *American Journal of Operations Research*, 03(01), 101–112.

Freeman, L.C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1), 35.

Herrera, M., Perez-Hernandez, M., Kumar Jain, A., and Kumar Parlikad, A. (2020). Critical link analysis of a national Internet backbone via dynamic perturbation. In *Advanced Maintenance Engineering, Services and Technologies*.

Kazil, J., Masad, D., and Crooks, A. (2020). *Utilizing Python for Agent-Based Modeling: The Mesa Framework*, volume 12268 LNCS. Springer International Publishing.

Kermarrec, A.M., Le Merrer, E., Sericola, B., and Trédan, G. (2011). Second order centrality: Distributed assessment of nodes criticity in complex networks. *Computer Communications*, 34(5), 619–628.

Lehmann, K.A. and Kaufmann, M. (2003). Decentralized algorithms for evaluating centrality in complex networks. *Networks*, (January 2003), 1–9.

Likic, V. and Shafi, K. (2018). Battlespace Mobile/Ad Hoc Communication Networks: Performance, Vulnerability and Resilience. 303–314.

Lü, L., Chen, D., Ren, X.L., Zhang, Q.M., Zhang, Y.C., and Zhou, T. (2016). Vital nodes identification in complex networks. *Physics Reports*, 650, 1–63.

Marsden, P.V. (2002). Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4), 407–422.

Moura, J. and Hutchison, D. (2019). Cyber-Physical Systems Resilience: State of the Art, Research Issues and Future Trends. 42.

Nanda, S. and Kotz, D. (2008). Localized Bridging Centrality for Distributed Network Analysis. In *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, 1–6. IEEE.

Neumann, P. (1995). *Fatal Defect: Chasing Killer Computer Bugs*, volume 20. Times Books, first edit edition.

Proselkov, Y., Herrera, M., Parlikad, A.K., and Brintrup, A. (2020). Distributed Dynamic Measures of Criticality for Telecommunication Networks. In *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, 1–12. Springer.

Salazar, J.C., Nejjari, F., Sarrate, R., Weber, P., and Theilliol, D. (2016). Reliability Importance Measures for Availability Enhancement in Drinking Water Networks. Technical report.

Veres, A. and Boda, M. (2005). *Complex Dynamics in Communication Networks*. Springer: Complexity.

von Mises, R. and Pollaczek-Geiringer, H. (1929). Praktische Verfahren der Gleichungsauflösung. *Zammzeitschrift Fur Angewandte Mathematik Und Mechanik*, 9, 152–164.

Wehmuth, K. and Ziviani, A. (2011). Distributed location of the critical nodes to network robustness based on spectral analysis. In *LANOMS 2011*, 1–8. IEEE.