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 maximise performance and resilience of daily operations. It requires large amounts of data delivered quickly. Data-transfer is enabled by telecom networks and network elements such as routers or switches. Disruptions can render a network inoperable, and advanced responsiveness to network usage is required to avoid them. This may be achieved by embedding autonomy into the network, providing fast and scalable algorithms that use key metrics for prioritising the management of a potential disruption, such as the impact of a failure in a network element on system functions. Centralised approaches are insufficient for this as they require time to transmit data to the controller, by which time it may have become irrelevant. Decentralised and information bounded measurements solve this by situating computational agents near the data source. We propose a method to assess the value of the amount of information for calculating decentralised criticality metrics. The method introduces an agent-based model that assigns a data collection agent to every network element and computes relevant indicators of the impact of a failure in a decentralised way. The method is evaluated through simulations of discrete information exchange and concurrent data analysis, comparing accuracy of simple measures to a benchmark, and computation time of the measures as a proxy for computation complexity. Results show relative 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 often represented by graphs, with network elements, such as routers or switches, as nodes and connections as edges.

In such flow networks, node failure may be caused by congestion at a node or edge (for example by data packets). Learning how to monitor the relative impact of disruptions such as congestion on the network - criticality - is important for network control. Accurate and quick control of network behaviour is important in networks that are functioning at or near capacity, as is expected for backbone networks of the near future (Moura and Hutchison (2019)).

In a centralised approach of 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)), and typically require node topology and attribute data to function. This must be live and dynamic to respond to behavioural shifts. The increased amount of data causes longer computational times, thus conclusions arrived at a given point in time become less relevant, and promote critical events if too much data is transferred. Therefore, the amount of data used for a CM should be minimised while preserving meaning. This may be achieved by imposing a limited bound around a given node, and computing criticality in a decentralised manner, also shortening data paths and reducing complexity by requiring 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)).

⋆ 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.

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 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 failure on the smooth operation of a network, evaluated by network connectivity in telecoms, is a key factor in understanding network resilience. One can determine criticality by tracking its impact on the network when rendered inactive (Lü et al. (2016); Herrera et al. (2020)), and estimate it using the current network state (Proselkov et al. (2020)). Criticality can inform prioritisation in network prognostics for proactive maintenance. Many criticality measures are extensions of 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 requires 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 packet exchange simulator is then run with short range dependence, meaning random nodes generate data packets independently according to a Poisson distribution with random destinations (Veres and Boda (2005)), using [1] a discrete-event network simulator for simulating networks that exchange discrete information packets. As this is an information flow network, a timestep refers to 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 some fixed capacity which gets filled up over time since it takes time to process packets at nodes and transmit them between nodes, with processing time and transmission time as fixed input parameters within the model. This process is terminated after either a fixed number of timesteps or until the network is too congested to function. The nodes all have a backlog capacity each of ϕ. 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 ϕ_u^t.

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

For a pair of connected 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, $(u, (u_i)_{i=0}^n, v) \subseteq V$ such that either all 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 receives information about ϕ_v^t. The queue at time t that the agent at node u believes v has is the *perceived queue*, $q_{uv}^t \in Q$. When $u = v$ this is q_u^t. Centrality calculation for nodes takes time so the time from transmission to output is always greater than transmission to receipt between any nodes, thus nodes must compute centralities at a lower frequency than the CP dictates to avoid losing currentness. We call the period between calculations the *monitor interval*, denoted μ.

Instant communication is a simplification that assumes monitoring data is transferred instantly without any lag, such that for all $u, v \in V$, $q_{uv}^t = q_v^t$. This is a base case, and

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

can only be achieved if monitoring data transfer became so fast as to be insignificant.

Constant communication has nodes declare their queues at every timestep and this data traverses the network normally, because this declaration is a very low bandwidth operation. If the shortest path (geodesic) between u and v is length n, then $q^t_{uv} = q^{t-n}_v$, since information from v takes n timesteps to reach u, so u's knowledge of v is n timesteps out of date. Due to this lag further nodes give less accurate information with reduced value and relevance.

Periodic communication has nodes declare their queues and perceived queues at the same frequency as they calculate their centrality. This corresponds to some aggregation of the functions that exist in the case of higher order control functions that use centralities as inputs. Here, for u and v with geodesic of length n, and model with monitor interval μ, $q^t_{uv} = q^{t-\mu n}_v$, as each queue pass is μ timesteps.

With each CP, nodes recieve 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

$$(uv)^t_q = q^t_u / |\Gamma_1(u)|. \tag{1}$$

This is done because larger neighbourhoods give nodes more opportunities 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 situtated 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 $H_i(u) = \bigcup_{j=1}^{i} \Gamma_j(u)$, such that 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 $C^{\mathrm{u}}_{\mathrm{d}}(u)_t = C^{\mathrm{u}}_{\mathrm{d}}(u) = |\Gamma_1(u)|$.

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

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

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

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

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

giving weighted sociocentric betweenness centrality as

$$C^{\mathrm{ws}}_{\mathrm{b}}(u)_t = \sum\nolimits_{v,w \in V, u \neq v \neq w} \omega^t_{v,w|u} / \omega^t_{v,w}. \tag{3}$$

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

Eigencentrality: *Unweighted Sociocentric Eigencentrality* captures the connectivity of the network, where nodes with more connections to well connected nodes are better valued. 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, λ_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 $A^t_G \in M_n(\mathbb{Z}^+)$,

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

A_G in Eqn. (4) is then replaced by A^t_G 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 estimate of criticality 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 and uses 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 window of timesteps, bisected by the present timestep, where the beginning of the window has sufficiently influenced all timesteps up to the present, and the present will sufficiently influence all timesteps up to the end of the window. With it, we can determine 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_{test}, of time horizon windows, h_i, less than half the total simulation runtime, $t_{\max}$, where $h_i = it_{\max}/(2h_{\mathrm{test}})$, and take moving averages over Φ^t for each width h_i, such that

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

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

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

and get the sum of absolute differences, $\mathrm{SAD}_i = \sum_t \mathrm{MAD}_i^t$. Normalised, this is $\mathrm{NSAD}_i = \mathrm{SAD}_i / \max_{i=1}^{h_{\mathrm{test}}} \mathrm{SAD}_i$. Iterating through NSAD_i in ascending i, we obtain $g_i = h_{\mathrm{test}}(\mathrm{NSAD}_{i+1} - \mathrm{NSAD}_i)$. 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:

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

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. This impact at time t is the change over h timesteps across t, scaled by the built up queues at time t, since a more 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, defined $\mathrm{THD}_t = \mathrm{MA}_{\Phi;h}^t \left(\mathrm{MA}_{\Phi;h}^{t+h} - \mathrm{MA}_{\Phi;h}^{t-h+1} \right)$. This is normalised to $[0,1]$, such that $\mathrm{NTHD}_t = (\mathrm{THD}_t - \min_t \mathrm{THD}_t)/(\max_t \mathrm{THD}_t - \min_t \mathrm{THD}_t)$. This is 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 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.

4. RESULTS AND DISCUSSION

We compared results of simulations of instant, constant, and periodic CPs for the relative 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).

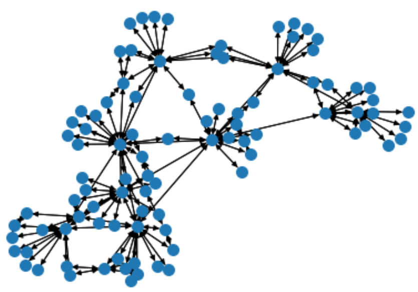


Fig. 1. Outer backbone UK infrastructure network for a large UK service provider

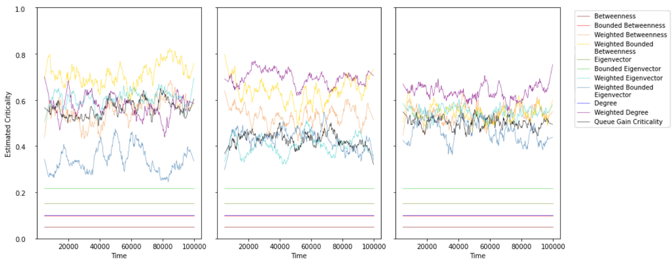
The parameters of the simulation are shown in Table 1. Since the simulation initialises on an empty network, we skip 5000 timesteps to avoid a degenerate case.

Table 1. Simulation Parameters

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

Manuscript 114 submitted to 14th IFAC Workshop on Intelligent Manufacturing Systems. Received October 15, 2021.

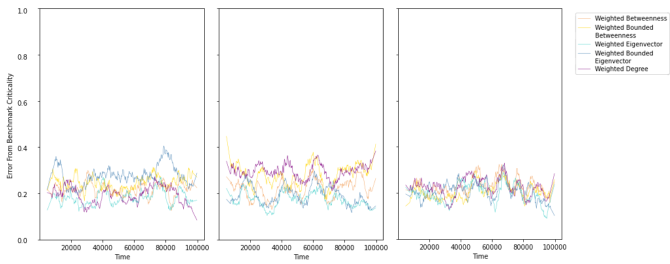
Plots of all centralities and the criticality benchmark, NTHD_t, for each CP are shown in Fig. 2, filtered using a first order Savitzky-Golay filter. This graph shows substantial difference between outputs for weighted and unweighted measures across all CPs, but further analysis will show similar accuracy.



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

Fig. 2. CEs and NTHD_ts for simulations of each CP.

The absolute error, $|\text{Err}_t|$, from NTHD_t was calculated, the results plotted in Fig. 3. These plots are only for the 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 seem 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. 3. Error for Weighted CEs and NTHD_ts for simulations of each CP.

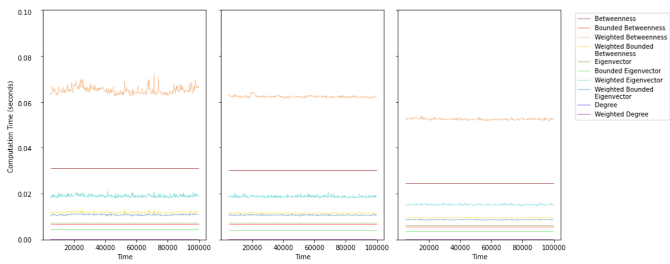
Difference from the criticality benchmark is shown in Table 2, with RMSE for each CE and each CP. Weighted, dynamic measures largely performed much better than their static counterparts. Boundedness minimally impacted accuracy. Typically, performance in estimating criticality was best for the constant CP, followed by the periodic and then instant CPs. Of the weighted bounded measures, betweenness performed best under periodic CP with large variation between CPs; eigenvector best under constant CPs; and degree under instant CP, and much better than periodic and constant, which show wide difference. This suggests different estimates can be used for different CPs. No weighted bounded measure had more RMSE than 0.35. Betweenness has the worst case in constant CP with RMSE 0.339, degree best case in instant CP with RMSE 0.239, and eigencentrality and betweenness showing similar consistency with ranges of 0.086 and 0.085 respectively. All values are similar and low, which suggests information bounding and dynamic measurement can be combined to create 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 maximise 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. We note that 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.

Computation time plots are displayed in Figure 4, where limiting information has the most noticeable effect on 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 computations are largely similar in time for all measures, though instant CP shows variability and intermittent spikes during network congestion, where queue backlogs grow due to build-up exceeding processing speed within 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 placed on the computer during simulation. Dynamicity increases computation time for complex measures, but has minimal impact on degree centrality, which is computed nearly instantly. The 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.

Altogether, this research shows the viability of dynamic criticality estimation and provides a framework for fur-

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

ther development of advanced CEs. Bounding information visibility is a viable method for scalable measures that largely preserves accuracy while speeding up calculation. The three main CPs examined were shown to behave similarly. Different measures respond differently to information limitation and dynamicity based on CP, suggesting further research into measure development is required.

This study simulated networks under normal operation. Further research will develop a framework to conduct analysis under critical event simulation, sourced from singular nodes. These scenarios are expected to introduce variability in criticality computation speed, which may have implications on the selection of monitoring intervals. Future research will also look at different network topologies for the purpose of generalising the relationship of accuracy and computation speed with the information boundary for different network topologies. This research used classic centrality measures for analysis purposes, and further research will used advanced measures developed to estimate criticality (Proselkov et al. (2020)), as well as produce case specific, machine learning derived CEs for maximum relevancy. 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.