# Toward Efficient Wide-Area Identification of Multiple Element Contingencies in Power Systems

Hao Huang
*Texas A&M University*
hao_huang@tamu.edu

Zeyu Mao
*Texas A&M University*
zeyumao2@tamu.edu

Mohammad Rasoul Narimani
*Texas A&M University*
narimani@tamu.edu

Katherine R. Davis
*Texas A&M University*
katedavis@tamu.edu



***Abstract*—Power system $N-x$ contingency analysis has inherent challenges due to its combinatorial characteristic where outages grow exponentially with the increase of $x$ and $N$. To address these challenges, this paper proposes a method that utilizes Line Outage Distribution Factors (LODFs) and group betweenness centrality to identify subsets of critical branches. The proposed LODF metrics are used to select the high-impact branches. Based on each selected branch, the approach constructs the subgraph with parameters of distance and search level, while using branches' LODF metrics as the weights. A key innovation of this work is the use of the distance and search level parameters, which allow the subgraph to identify the most coupled critical elements that may be far away from a selected branch. The proposed approach is validated using the 200- and 500-bus test cases, and results show that the proposed approach can identify multiple N-x contingencies that cause violations.**

***Index Terms*—Power system contingency analysis, line outage distribution factors, graph theory, group betweenness centrality.**

## I. Introduction

As a reliability requirement, modern power grids must have the ability to withstand $N-1$ contingencies. However, $N-1$ analysis fails to capture high-impact scenarios due to increasing threats from cyber and physical domains that can cause multiple elements to disfunction or malfunction concurrently, potentially leading to cascading failures in the system and large-scale blackouts. as evident from recent examples [1]–[3]. The well-known Northeast blackout in 2003 affected 55 million people, and was caused by cascading failure. A cascading failure refers to a sequence of dependent events, where the initial failure of one or more components trigger the sequential failure of other components [4], [5]. Identifying and protecting the critical components that can trigger a cascading failure is an important need that would enable grid operators to prevent cascading failures and operate the system reliably.

Contingency analysis, a key problem in power system operation, is a systematic study of the impact of an individual or a group of system component failures on the overall system [6]. In general, $N-x$ contingency analysis, where $x \geqslant 2$, studies the impact of various combinations of $x$ individual components failing concurrently [7]. The number of multiple components assessed in a $N-x$ contingency analysis grows exponentially with $x$. For instance, the number of contingency cases for $N-1$ analysis is 20000 for a system with 20000 components, while the number of contingency cases for $N-2$ and $N-3$ analyses are approximately $10^8$ and $10^{12}$; this clearly becomes intractable as $x$ increases [7].

To identify the influence of an element in a network, numerous studies have utilized different variations of centrality metrics in graph theory, including betweenness centrality, closeness centrality, graph centrality, stress centrality, degree centrality, and more. The approach in [7] modifies the betweenness centrality metric to identify multiple critical components whose loss can trigger cascading failures. The betweenness centrality metric is employed to identify the most critical component in power grids [8], [9]. The study in [10] applies the graph edge betweenness centrality metric to identify critical components in the large-scale power systems. In addition, various electrical properties have been proposed to be considered together with centrality metrics to increase the accuracy of critical component identification, such as the admittance matrix [11], electrical distance [12], and the maximal load demand and the capacity of generators [13] to identify critical elements in power systems.

A key important factor in identifying critical components in electric power grids is the impact that the loss of components might have on the system operation. None of the above studies consider the impact of component loss in identifying critical components. Our previous work [14] applies Line Outage Distribution Factors (LODFs) [15] and betweenness centrality metrics to identify multiple critical branches in electric power grids. In this work, we improve upon the approach in [14] to search in a wider area in the corresponding graph of the electric power grid, which enables the method to evaluate coupled critical branches that may be far away from each other. These geographically distributed coupled elements may be missed in the previous approach, and they are important to identify. Unlike the previous approach, which constructs the subgraph only with nearby branches, this work introduces a new parameter *distance* that enlarges the searching graph. From the numerical results, the new approach can find more subsets of critical branches that cause more severe contingencies. The main contributions of this paper are thus as follows:

1) A new parameter *distance* is introduced to enlarge the subgraph for the group betweenness centrality approach to identify critical branches.
2) The resulting method is applied and evaluated on 200-bus and 500-bus synthetic grids, and the method's *distance* and *searching level* parameters are varied.
3) From the contingency analysis results and comparison with [14], the new approach can find more subsets of critical branches causing more violations.

This paper is organized as follows. Section II reviews the method in [14] on how to utilize LODFs and group betweenness centrality to find critical branches in large scale power grids. Section III presents the improved approach for

finding the most critical branches in a wider area. Section IV empirically evaluates the proposed approach in 200- and 500-bus synthetic power grids. Section V concludes the paper and discusses future work.

## II. Related Work

This section reviews how LODFs and group centrality betweenness are utilized in the method proposed by the authors in [14] to identify critical multiple element contingencies. This method is a precursor to the extension developed in this paper.

### A. Line Outage Distribution Factors

To incorporate LODFs as a metric to identify the importance of a selected branch, in the method proposed in [14], the mean of the absolute value of the remaining branches' LODFs after the selected branch's outage is normalized with the standard deviation. Equation (1a) shows the metric $NLODF(i)$ based on normalized absolute values of LODFs,

$$NLODF(i) = \frac{mean(abs(LODFs))}{std(abs(LODFs))} \tag{1a}$$

$$M(i) = PF(i) \times min\{NLODF(i), 1\} \tag{1b}$$

where $PF(i)$ is the power flow in line $i$ during the normal operation; $mean\,(\cdot)$, $std\,(\cdot)$, and $abs\,(\cdot)$ indicate the mean, the standard deviation, and the absolute functions, respectively. The $min$ function in (1b) enforces that $NLODF(i)$ is less than or equal to one, e.g., when an islanding situation is encountered.

### B. Group Betweenness Centrality

The betweenness centrality for an element in the graph can be defined as the frequency at which that element (i.e., a node or edge) is in the shortest path between the node pairs of the entire graph. To apply the betweenness centrality into $N-x$ contingency analysis to identify multiple branches in a graph, the method in [14] extends the betweenness centrality metric to the group betweenness centality (GBC) metric. The method can then be utilized to identify multiple critical branches simultaneously in a graph. The GBC metric is mathematically represented as follows:

$$GBC(E) = \sum_{s=1}^{n}\sum_{t=1}^{n}\frac{\sigma(s,t|E)}{\sigma(s,t)}, s,t \notin E, s \neq t \tag{2}$$

The $E$ in equation (2) represents the subset of edges of interest, $\sigma(s,t)$ is the number of shortest paths between $s$ and $t$, and $\sigma(s,t|E)$ is the number of shortest paths between $s$ and $t$ that contain any element in $E$.

## III. The Framework of Identifying Critical Branches Over the Whole Grid

The proposed method extends upon [14] that constructs the searching graph using *searching level* ($sl$), which only considers branches that are around the desired branch. In this paper, we enlarge the searching graph by looking at branches that are far away from the desired branch. Thus, we introduce a new parameter, *distance* ($d$), that quantifies the distance between the branches with the highest $M$ value and other desired branches of interest. The $sl$ parameter will then define a larger searching graph based on the branches of interest. Then, by using the $GBC$ and $LODFs$, the proposed approach identifies the critical multiple-element branch contingencies in a wider area.

Figure 1 summarizes the multiple steps of the proposed method for identifying critical branches. First, based on the system information, we compute the $NLODF$ and $M$ for each line. Then, we select the first $a\%$ of the branches with the highest value as the starting point to construct each subgraph. The $d$ determines the distance between the line with the highest $M$ value and other desired branches of interest. For instance, for $N-3$ contingency analysis, we first select the line with the highest $M$ value and then find the branches with high $M$ in the vicinity of the first selected line within $d$-hop distance from the first selected line. Once these branches are determined, we use $sl$ to find the nodes that are within the $sl$-hop distance from both ends of the selected branches. Note that $sl \geqslant d$ in order to guarantee the connectivity of the sub-graph. All these nodes create a graph which is a sub-graph of the underlying graph of the test case. At last, we apply the $GBC$ for each subgraph and identify the $X$ most critical branches in each subgraph.

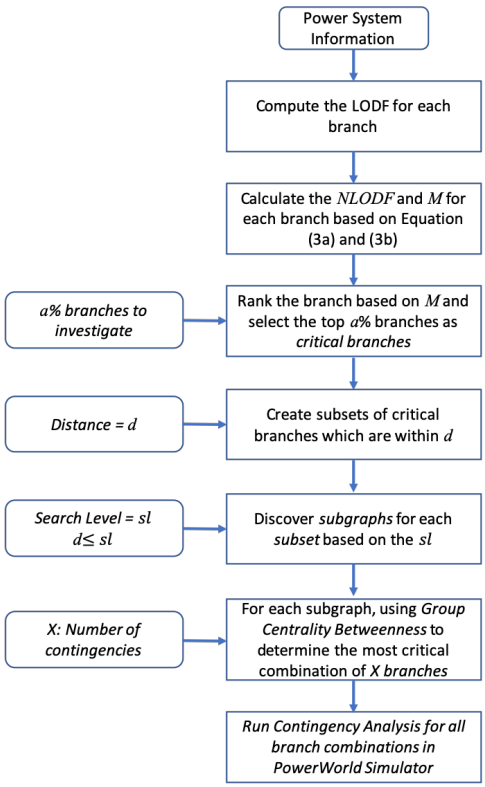


Figure 1. Critical Branches Identification Framework

To construct the subgraph, the proposed approach first selects the branches with highest $M$, which is shown with green star ends in Figure 2. Based on the $d$ parameter, the method then selects the neighboring branches that have high $M$ and are within $d$-hop distance, which are shown with yellow diamond ends in Figure 2. The green star nodes and yellow diamond nodes constitute the desired branches, which are the bone nodes in subgraph. Then, the subgraph selects

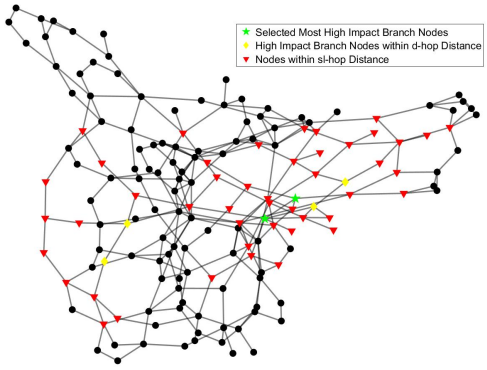


Figure 2. The equivalent graph of the pglib_opf_case162_ieee_dtc test cases [16] with the $d$ = 3 and $sl$ = 3. The green star nodes show both ends of the line whose outage has the highest $M$. The yellow diamond nodes show other high impact branches in the grid that are within 3 hop-distance to the green star nodes. The red triangle nodes are within 3-hop distance from the desired branches (green star nodes and yellow diamond nodes).

branches that are within $sl$-hop distance from the ends of these desired branches, which are shown with red triangle ends in Figure 2. All colored nodes in Figure 2 are the subgraph for one of the first $a\%$ of the most impactful branches in the grid.

## IV. Numerical Results

In this section, we apply our approach to two synthetic test cases, a 200-bus and 500-bus case respectively, from the benchmark library for electric power grids in Texas A&M University [17]. We implement our approach in Python and use ESA [18] to communicate with PowerWorld Simulator to collect LODFs. The results are computed using a laptop with an i7 1.80 GHz processor and 16 GB of RAM.

### ■ 200-Bus Test System

This 200-bus synthetic test is a relatively small size test system system with 245 branches and 49 generators [17] and it is selected to evaluate the effectiveness of the proposed method. Both brute force and the proposed contingency analysis methods have been applied in this case and find the same critical lines for the $N-1$ and $N-2$ contingency analysis. Comparing the computational time of both methods for $N-2$ contingency analysis, the proposed algorithm can find the result within 100 seconds for $sl = 3$, while it took 230 seconds for the brute force search. For $x > 2$, the brute force method can be hardly applied. This makes the proposed approach a good candidate to perform a higher order $N-x$ contingency analysis in larger test cases where it is not possible to find critical lines by the brute force search method.

The proposed algorithm is utilized to solve different levels of the $N-x$ contingency analysis for 200-bus test systems. The contingency analysis for different combinations of $d$ and $sl$ are solved to evaluate the impact of these parameters on finding the critical branches in the underlying test system. The results for 200-bus test system for $sl = 4$ and $d = 4$ are tabulated in Table I. The first column in this table lists the order of the contingency analysis (i.e., $x$ in the $N-x$ term). The second and the third columns represent critical branches and contingency violation types, respectively. Various types of limit violations, including *reserve limit*, *overflow*, *undervoltage*, and *unsolved* are considered in this paper. Note

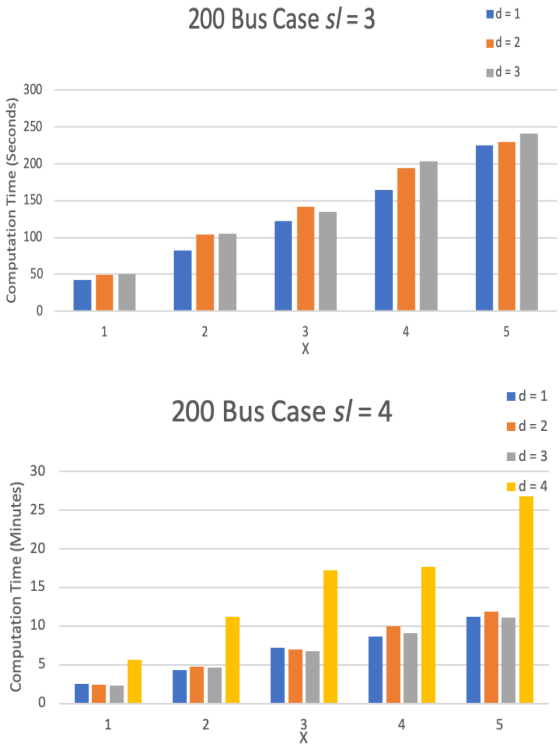


Figure 3. Computation Time for 200 Bus Case Against Different *Search Level*, *Distance* and *X*

that the *unsolved* case mirrors the situation where there is no solution for the power flow equations. The types of contingency violations in the third column are found via removing the listed critical branches in the second column. The proposed algorithm can find more critical branches with more severe contingencies for 200-bus test system compare to the approach in [14], which are highlighted in Table I. Finding new critical branches by the proposed algorithm authenticate its superiority on our previous approach. The one-line diagram of the 200-bus test cases and the corresponding violations caused by the outage of branches [136, 133], [135, 133], and [125, 123] is depicted in Figure 4.

The execution time of the proposed algorithm for various contingency levels ($x$) and different combinations of $sl$ and $d$ are visualized in Figure 3. Figure 3 shows that the execution time for a specific contingency level often linearly increases by $d$ increment. The important point is that the execution time of contingency analysis for specific search level and distance increases linearly as $x$ increases. This characteristic qualifies the proposed approach for performing different levels of contingency analysis in larger test cases where the problem is computationally intractable for traditional contingency analysis methods.

### ■ 500-Bus Test System

With 597 branches and 90 generators, the 500-bus test system can challenge the ability of different approaches in identifying critical branches in electric power grids.

The 500-bus test system is resilient enough that it is hard to identify a limit violation even by randomly removing multiple branches. Nevertheless, the proposed algorithm easily identifies multiple limit violations caused by outage of few

Table I
RESULTS FROM APPLYING THE PROPOSED APPROACH TO 200-BUS TEST SYSTEM WITH $d = 4$ AND $sl = 4$.

| X | Critical Line | Violations |
|---|---|---|
| 1 | [189, 187] | Reserve Limit |
| 2 | [189, 187], [187, 121] | Reserve limit |
| 2 | [189, 187], [136,133] | Reserve limit |
| 2 | [136, 133], [135, 133] | 1 Overflow and Reserve limit |
| 3 | [189, 187], [187, 121], [154, 149] | Reserve Limit |
| 3 | [189,187], [136, 133], [135, 133] | Unsolved |
| 3 | [136, 133], [135, 133], [125, 123] | 2 Overflow, 18 Undervoltage and Reserve Limit |
| 4 | [189, 187], [136, 133], [135, 133], [125, 123] | Unsolved |
| 4 | [189, 187], [187, 121], [154, 149], [152, 149] | 2 Overflow |
| 5 | [189, 187], [136, 133], [135, 133], [125, 123], [126, 123] | Unsolved |
| 5 | [189, 187], [187, 121], [154, 149], [152, 149], [153, 149] | Unsolved |
| 5 | [136, 133], [135, 133], [125, 123], [126, 123], [127, 123] | Unsolved |

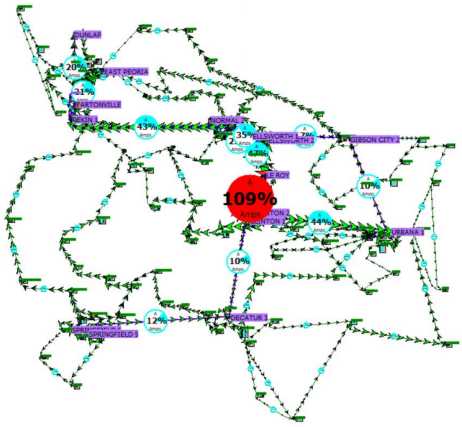


Figure 4. 200-bus Test System after Outages of Following Branches [136,133][135,133][125,123]

branches. For instance, the outage of the following branches, i.e. [162, 220], [23, 386], and [87, 141], cause an overflow violation in the system. Multiple limit violations caused by few line outages are listed in Table II and Table III for different values of $sl$ and $d$. Although the 500-bus test system is resilient, the proposed approach is able to find multiple limit violations caused by three to five line outages. This verifies the ability of the proposed approach in solving contingency analysis in relatively large test systems.

Compare to our previous approach in [14], the proposed approach identifies new critical branches in 500-bus test system, which are listed in highlighted rows in Table II and Table III. This verifies that the proposed algorithm can search for critical branches in the system more efficiently compare to the previous approach in [14].

The computational time for solving different levels of contingency analysis problems for 500-bus test system for various $d$ and $sl$ are shown in Figure 5. Similar to computational times of contingency analysis in 200-bus test system, the computational time of contingency analysis for 500-bus test system linearly increases by $d$ increment for specific $sl$ and contingency analysis level, i.e. $x$. Comparing the the plots in Figure 5 shows that the computational time is more sensitive to the $sl$ rather than the $d$. It is because the number of neighboring nodes (i.e. red triangle nodes in Figure 2) that needs to be evaluated increases by the search level value. However, the computational time increases linearly with increment in $sl$ and

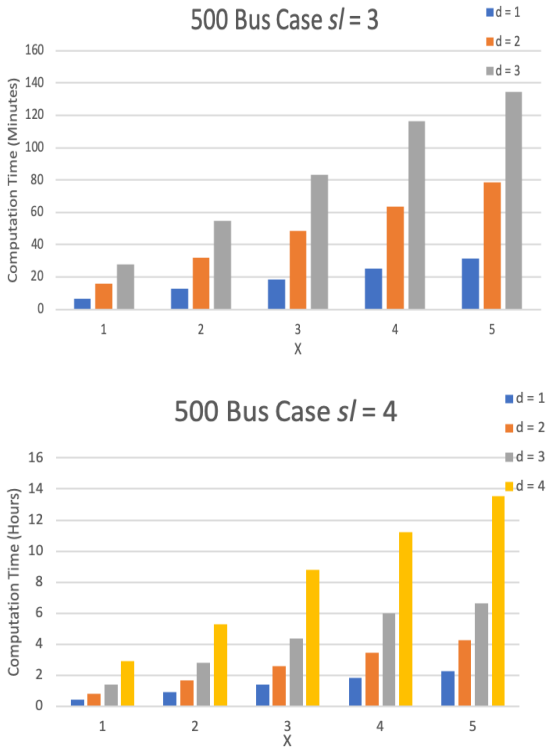


Figure 5. Computation Time for 500 Bus Case Against Different *Search Level*, *Distance* and *X*

$d$, which makes the proposed approach tractable for rendering contingency analysis in large test systems.

## V. CONCLUSION AND FUTURE WORK

The proposed approach provides a computationally tractable approach to identify critical multiple-element branch contingencies by exploiting the group betweenness centrality and LODFs. The capability of the proposed method in finding critical branches in electric power grids is examined by two synthetic grids and different $N - x$ contingency analyses in these systems. The obtained results demonstrate that the proposed method computes the high-impact multiple-element contingencies in realistic synthetic test systems. The proposed approach can help power system operators to identify critical branches and make proper decisions to preserve power system

Table II
RESULTS FROM APPLYING THE PROPOSED APPROACH TO 500-BUS TEST SYSTEM WITH $d = 2$ AND $sl = 3$.

| x | Critical Line | Violations |
|---|---|---|
| 3 | [142, 141], [424, 423], [87, 141] | 3 Overflow |
| 3 | [162, 220], [23, 386], [87, 141] | 1 Overflow |
| 4 | [162, 220], [23, 386], [87, 141], [247, 246] | Unsolved |
| 4 | [142, 141], [424, 423], [87, 141], [247, 246] | Unsolved |
| 5 | [162, 220], [23, 386], [87, 141], [247, 246], [437, 428] | Unsolved |
| 5 | [162, 220], [23, 386], [142, 141], [424, 423], [87, 141] | 5 Overflow |
| 5 | [142, 141], [424, 423], [87, 141], [247, 246], [402, 401] | Unsolved |

Table III
RESULTS FROM APPLYING THE PROPOSED APPROACH TO 500-BUS TEST SYSTEM WITH $d = 2$ AND $sl = 4$.

| X | Critical Line | Violations |
|---|---|---|
| 3 | [142, 141], [424, 423], [87, 141] | 3 Overflow |
| 4 | [162, 220], [23, 386], [87, 141], [247, 246] | Unsolved |
| 5 | [142, 141], [424, 423], [87, 141], [247, 246], [402, 401] | Unsolved |
| 5 | [268, 267], [213, 212], [105, 104], [408, 407], [36, 35]] | 2 Overvoltage |

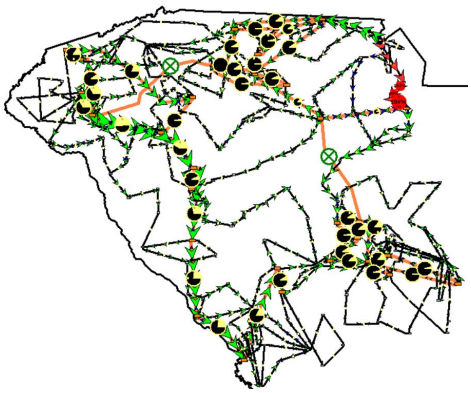

Figure 6. 500-bus Test System after Outage of the Following Branches [162, 220][23, 386][142, 141][424, 423][87, 141]

reliability via protecting these critical branches against natural incidents, cyber-attacks, etc.

For future work, there are two potential ways to improve the framework's efficiency and speed. First, from the results on computation times, it is obvious that the required computation time increases as the subgraph size increases. However, a larger subgraph is not guaranteed to identify all critical branches. It is more efficient to create subgraphs without too much overlapping. Obtaining the appropriate parameter pair of $d$ and $sl$ through analyzing subgraphs' overlapping situation for each case can improve the overall efficiency. Secondly, the proposed method is searching critical branches repeatedly over each subgraph. Thus, applying the framework with parallel computing can improve its speed greatly, which can make it more applicable in industry.

## VI. ACKNOWLEDGEMENT

The work described in this paper was supported by funds from the US Department of Energy under award DE-OE0000895 and the National Science Foundation under Grant 1916142.

## REFERENCES

[1] M. Sahraei-Ardakani, X. Li, P. Balasubramanian, K. W. Hedman, and M. Abdi-Khorsand, "Real-time contingency analysis with transmission switching on real power system data," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2501–2502, 2016.

[2] X. Li, P. Balasubramanian, M. Sahraei-Ardakani, M. Abdi-Khorsand, K. W. Hedman, and R. Podmore, "Real-time contingency analysis with corrective transmission switching," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2604–2617, 2017.

[3] A. Bauman, "Nyc blackout: Cause of massive manhattan outage under investigation," jul 2019.

[4] Y. Zhu, J. Yan, Y. Sun, and H. He, "Revealing cascading failure vulnerability in power grids using risk-graph," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3274–3284, 2014.

[5] Vaiman, Bell, Chen, Chowdhury, Dobson, Hines, Papic, Miller, and Zhang, "Risk assessment of cascading outages: Methodologies and challenges," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 631–641, 2012.

[6] North American Electric Reliability Corporation, "Reliability guideline methods for estabilishing irols," Sep 2018.

[7] M. Halappanavar, Y. Chen, R. Adolf, D. Haglin, Z. Huang, and M. Rice, "Towards efficient n-x contingency selection using group betweenness centrality," pp. 273–282, Nov 2012.

[8] Z. Wang, A. Scaglione, and R. J. Thomas, "Electrical centrality measures for electric power grid vulnerability analysis," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5792–5797, Dec 2010.

[9] E. P. R. Coelho, M. H. M. Paiva, M. E. V. Segatto, and G. Caporossi, "A new approach for contingency analysis based on centrality measures," *IEEE Systems Journal*, vol. 13, pp. 1915–1923, June 2019.

[10] I. Gorton, Z. Huang, Y. Chen, B. Kalahar, S. Jin, D. Chavarría-Miranda, D. Baxter, and J. Feo, "A high-performance hybrid computing approach to massive contingency analysis in the power grid," in *2009 Fifth IEEE International Conference on e-Science*, pp. 277–283, IEEE, 2009.

[11] G. chen, Z. Yang Dong, D. J. Hill, and G. Hua Zhang, "An improved model for structural vulnerability analysis of power networks," *PhysicaA*, vol. 388, no. 23, pp. 4259–4266, 2009.

[12] E. Bompard, D. Wu, and F. Xue, "Structural vulnerability of power systems: A topological approach," *Electric Power Systems Research*, vol. 81, no. 7, pp. 1334 – 1340, 2011.

[13] K. Wang, B. Zhang, Z. Zhang, X. Yin, and B. Wang, "Hybrid flow betweenness approach for identification of vulnerable line in power system," *PhysicaA*, vol. 390, no. 23, pp. 4692–4701, 2011.

[14] M. R. Narimani, H. Hao, A. Umunnakwe, Z. Mao, A. Sahu, S. Zonouz, and K. R. Davis, "Generalized contingency analysis based on graph theory and line outage distribution factor," *arXiv:2007.07009*, July 2020.

[15] C. M. Davis and T. J. Overbye, "Multiple element contingency screening," *IEEE Transactions on Power Systems*, vol. 26, pp. 1294–1301, Aug 2011.

[16] IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms, "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms," *arXiv:1908.02788*, Aug. 2019.

[17] benchmark library for electric power grids in Texas A&M University, available at https://electricgrids.engr.tamu.edu

[18] B. L. Thayer, Z. Mao, Y. Liu, K. Davis, and T. Overbye, "Easy simauto (esa): A python package that simplifies interacting with powerworld

simulator," *Journal of Open Source Software*, vol. 5, no. 50, p. 2289, 2020.