HotSpots: Failure Cascades on Heterogeneous Critical Infrastructure Networks

Liangzhe Chen*, Xinfeng Xu*,°, Sangkeun Lee+, Sisi Duan+
Alfonso G. Tarditi+, Supriya Chinthavali+, B. Aditya Prakash*
*Department of Computer Science, Virginia Tech
°Department of Physics, Virginia Tech
+Oak Ridge National Laboratory
{liangzhe, badityap}@cs.vt.edu, xinfeng@vt.edu, {lees4, duans, tarditiag, chinthavalis}@ornl.gov

ABSTRACT

Critical Infrastructure Systems such as transportation, water and power grid systems are vital to our national security, economy, and public safety. Recent events, like the 2012 hurricane Sandy, show how the interdependencies among different CI networks lead to catastrophic failures among the whole system. Hence, analyzing these CI networks, and modeling failure cascades on them becomes a very important problem.

However, traditional models either do not take multiple CIs or the dynamics of the system into account, or model it simplistically. In this paper, we study this problem using a heterogeneous network viewpoint. We first construct heterogeneous CI networks with multiple components using national-level datasets. Then we study novel failure maximization problems on these networks, to compute critical nodes in such systems. We then provide HotSpots, a scalable and effective algorithm for these problems, based on careful transformations. Finally, we conduct extensive experiments on real CIS data from multiple US states, and show that our method HotSpots outperforms non-trivial baselines, gives meaningful results and that our approach gives immediate benefits in providing situational-awareness during large-scale failures.

1 INTRODUCTION

Modern critical infrastructures (CIs) such as Energy, Water, Communication etc are mutually dependent in such complex ways. For example, the energy network depends on the water network for treatment, dissemination, and disposition and the water network relies on the energy network for energy production [28]. Indeed, such dependencies exist across multiple CIs, where hazards/failures affecting one CI network can potentially propagate to other networks and disrupt the functionality of the entire system.

The 2003 Northeastern US blackout [5] is a perfect example depicting the risk, where a fault in 3 transmission lines caused a massive blackout impacting multiple CIs. Initially, the massive power outage caused blackouts across several U.S. states. The massive power outage effects cascaded to drinking and waste-water treatment systems, communication, transportation, and a number of major business and food services. Nearly 50 million people were affected, causing huge economic losses exceeding $5 billion. About every 4 months in the US, a major blackout occurs, affecting one million or more people [24]. This increased vulnerability of single CI network can be easily amplified due to the interdependencies. Similar examples abound such as in Hurricane Sandy where cascading and escalating failures caused severe impacts to the infrastructures and slowed down the recovery tremendously [10]. Hence, modeling and simulation of interdependent CI systems (CISs) has become an important field to realize the idea of 'smart cities and nations'.

Since the 2003 NE blackout, FEMA (Federal Emergency Management Agency), other federal agencies such as DOE-OE, DoD and several national labs are constantly working towards improving wide-area situational awareness and developing sophisticated decision support tools that can help in predicting propagating impacts due to an extreme event like hurricanes, wildfires etc [2]. For e.g, Oak Ridge National Lab (ORNL) developed detailed hurricane outage models to predict which substations will be impacted, and what the downstream implications of these failures are even before a hurricane hits the land [8]. Such situational awareness can assist mitigation planning with the available resources, and support the construction of more resilient systems for future.

To this end, identifying critical and vulnerable nodes and links of these interconnected networks that can cause maximum damage is very important so that the system reliability and efficiency can be improved by monitoring and protecting them [22, 26]. For example, based on topological analysis of the NE American grid, it was identified that nodes with a high degree are more important than others [6]. Centrality indexes based on betweenness were defined by several researchers to analyze the vulnerability of power systems [19]. Similarly from a data-mining view point, a few recent studies have tried to model failures in infrastructure networks [13, 14]. However, all these methods either work single CIs, or do not take into account any dynamics of the system, or model it very simplistically (like considering just one-step failures).

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3132867

In this paper, a collaboration between computer scientists and power engineers, we broadly approach this problem using heterogeneous networks. We unify various CI systems by first constructing a 5-component heterogeneous network constructed by combining various individual CIs. Looking at the system as a heterogeneous network gives many advantages. We then study the problem of finding the k critical nodes (the 'hot spots') in the power grid network, the failure of which would cause the maximum failures across the CI networks. We design a novel and an effective way of predicting non-linear interaction among the nodes the result of which may not be visible through a static structural analysis.

Our contributions are:

(1) We construct heterogeneous networks from real CI datasets and develop a novel tractable cascade model F-Cas.
(2) We develop an efficient and effective algorithm HotSpots to find the most critical nodes in the CIS using F-Cas, whose failure can cause maximum damage.
(3) Through our extensive experiments and case studies on unique large real datasets at ORNL, we show that our approach has immediate benefits to CI analysis, that HotSpots outperforms non-trivial baselines and effectively finds the vulnerable nodes and the results from HotSpots helps with real world situations.

2 OUR SETUP AND FORMULATIONS

Here, we first introduce the five CI components we consider in this work and how we interlink them to a heterogeneous network G. Second, we propose a failure cascade model F-Cas on G that captures the interdependencies between different components in G. And finally we formally define our problems.

Preliminary: IC Model. Given a weighted directed network, the popular Independent Cascade (IC) model [20] describes the spread of a contagion (idea/influence etc.) over it. Once infected/activated, each node gets one chance to activate its neighbors in the next time step with probability equal to the connecting edge-weight. The cascading process starts with an initial active 'seed' set, and ends when there is no new activation.

2.1 Network Construction

Interpretation and conversion of a set of GIS (geographic information system) datasets into a heterogeneous network is a crucial but challenging task. With no systematic tools available, researchers need to understand geographical objects and their relationships carefully, and write conversion scripts for every different analytic purposes. To avoid such ad hoc proprietary data processing, we develop and utilize a generic reusable urbannet-toolkit to systematically construct CI heterogeneous networks for our analysis.

The urbannet-toolkit we design contains four components. **shp2csv** converts shapefile data (a prevalent data format used in GIS which is not suitable for network analysis) to csv files; **csv2net** constructs node lists or edge lists from different shapes (POINT, MULTIPOLYGON, MULTILINESTRING); if needed, **net-simplifier** simplifies the networks by removing redundant nodes that are used to depict the shape contour; and finally, **net-linker** interconnects different CI networks based on user-specific criteria.

Specifically we select five important components from the HSIP Gold data [1] and EIA data [3] from the power system, and the natural gas system (as shown in Tab. 1). Among them, power plants, substations, and natural gas compressors are facilities without connections among themselves. For example, a power plant do not directly connect to another power plant, the connections are through other types of facilities. While in the transmission network and the pipeline network, each node represents a connection point between two transmission lines or pipelines, and the link between two nodes represents the actual transmission lines/pipelines. These components contain a natural support chain, where the power plants use the natural gas as fuel to generate electric power, the transmission nodes deliver the power to substations, the substations distribute power to natural gas compressors, and finally natural gas compressors help deliver natural gas to power plants through the pipelines. Note that there are different types of power plants which use different fuels, here we only consider those which use natural gas as fuel.

Infrastructure Type	Node Type	Description
Power	Electrical power plants (g)	Generate electrical power which is transmitted to substations through the transmission network.
	Transmission nodes (t)	Move electrical power from power plants to substations.
	Electrical substations (s)	Transform voltage and distribute electrical powers to consumers.
Natural gas	Natural gas compressors (c)	Increase the pressure of a gas to transport it through pipelines.
	Pipelines (p)	Transport natural gas to consumers.

Table 1: Summary of the five components in G.

To realize such a support chain in the system, we create interlinks between different components in the following ways.

Substations are connected to the nearest transmission node since it gets electrical supply from it. Each substation is also connected to the natural gas compressors within its service area to capture the fact that it provides power to these local facilities (service areas are non-overlapping, so each natural gas compressor is connected to only one substation).

Power plants are connected to the nearest natural gas pipeline and transmission node, since they get fuel from the pipelines and output power through the transmission network.

Natural gas compressors are connected to the nearest pipeline to capture the fact that the flow and the pressure of the natural gas along these pipelines depends on the compressors.

We summarize the network structure in Fig. 1. Our final directed heterogeneous graph is $G(V, E)$, where $V = \{V_g, V_t, V_s, V_c, V_p\}$ contains all nodes in the five CI components; and $E = \{E_t, E_p, E_{inter}\}$ contains the edges in the transmission network, pipeline network, and all the interlinks we created above. The directions of edges are either indicated from the data themselves, or from the feed-supply relation we introduce for creating the interlinks.

2.2 Failure Cascade Model F-Cas

The CI network system is vulnerable to potential failure cascades, as localized failures may get amplified to system-wide levels. For example, the failure of nodes in the transmission network would force

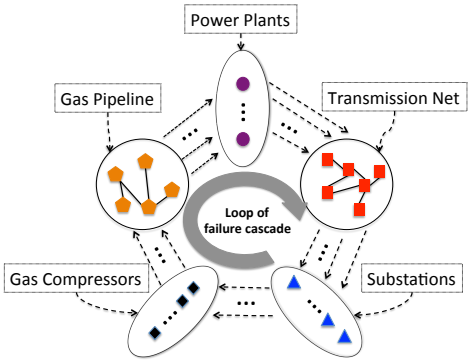


Figure 1: Interconnections and structure of G.

re-routing of the power flow and cause overload, or near-capacity operations of other transmission lines, with an increased probability of failure of additional nodes. With widespread, scattered, power interruptions, natural gas compressors may eventually lose power, thus affecting gas-fueled power plants, further reducing the overall generation available to support the load.

Such failures are very hard to model due to the complicated physical equations involved: state-of-the-art supercomputers utilizing sophisticated parallelization algorithms require *tens of days* to perform accurate simulations just for the power system[1]. These are intractable for analysis and also hard to interface with other tools. Hence we propose a failure cascade model F-Cas which makes simplified yet realistic enough assumptions. The goal and the challenge is to capture the most important unique dynamics quickly, to *estimate* failure cascades, and then extract critical nodes. See Alg. 1. Overall, we first simulate the failure cascade caused by overloads within the transmission network. Based on such failures, we detect, for each other type of node, if they would fail until there are no new failures (this process may continue for several iterations or 'loops' until convergence). Next, we define the failure conditions for each CI component in F-Cas.

Substations fail when they have no path in the transmission network to an active power plant, due to the lack of power.

Natural gas compressors fail when their associated substation (from which it gains power) fails.

Power plants fail when any of the natural gas compressors it connects to fails, due to the lack of fuel.

Pipelines serve as the connection between natural gas compressors and the power plants, they do not depend on other facilities and hence we assume they would not fail during the failure cascading. In reality, if some pipelines are damaged by some natural disaster, they can be removed from our analysis in advance.

Transmission nodes may fail due to any overload resulting from power re-routing in the transmission network caused by the failure of other transmission nodes (the power has to go through other routes which increases the probability of an overload of other transmission nodes). To capture such failure cascades due to overloading, we propose two Independent Cascade (IC) style models.

Trans-naive: When a transmission node fails, its children in the transmission network (the nodes that consume power from it) would have to gain power from other nodes. This increases the chance of overloading of other nodes. So in this model, we simply assume that when a node fails, its '*co-parents*' (nodes that share a common child node) have a certain probability to overload. Using this assumption, we first identify the co-parent nodes according to the transmission network, and then create a new *co-parent* network (where two transmission nodes are connected if they are co-parents in the original network). The edge-weights will be:

$$e_{ij} = \begin{cases} c & \text{if } t_i \text{ and } t_j \text{ share a child} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where c is some constant probability/weight, which represents the probability of a node failing its co-parents. The failure cascade process in the transmission network can be thought of as an IC model on a co-parent transmission network.

Trans-real: We make a more realistic assumption here: the probability of a transmission node's failure would be higher when the loading operating condition (e.g. power transferred) is closer to an established engineering limit (here referred to as the node power capacity) [29]. Based on this assumption, we design the following edge weights for all edges in the transmission network, representing the probability of node t_j's failure given the failure of its parent node t_i:

$$e_{ij} = \frac{\sum_{x \in Cs(Par(t_j) \setminus t_i)} Load(x)}{\sum_{x \in Par(t_j) \setminus t_i} Capacity(x)} \quad (2)$$

where $Par(t_j)$ are the parents of t_j, $Cs(Par(\cdot))$ are the union of the child nodes of parent nodes in $Par(\cdot)$. Note that unlike Trans-naive, the above edge weights are defined directly on the edges in the transmission network instead of an additional co-parent network. Basically we look at the parents of t_j (except t_i), and calculate the ratio of the total load from their children, with the capacity the parents have. If the ratio is closer to 1, the operating loads of the parents are closer to their engineering limits, and hence it is more likely that they fail to provide enough power to t_j.

Novelty: Note that our final failure cascade model F-Cas is a combination of all the different types of failures mentioned above, which cannot be simply represented by popular cascade-style models in standard literature [15, 20]. For example, consider the the path-based failure condition of a substation. In typical cascade models such IC, the failures are passed from one failed neighbor to another (through the connecting edge). However in F-Cas, a substation may fail due to a *non-local* failure of some transmission node which may be far away. Hence the typical influence-analysis algorithms cannot be directly applied for our problem, and novel techniques are needed.

Algorithm 1 A simulation process for F-Cas

Input: G, a seed node set S, Trans-real (or Trans-naive)
Output: The set of failed nodes W.

1: $W = S$
2: Run Trans-real/Trans-naive simulation to find the set of transmission nodes that fail (add them to W).
3: **while** $|W|$ is increasing **do**
4: **for** each substation $s \notin W$, compressor $c \notin W$, and power plant $g \notin W$ **do**
5: Check if it fails according to its failure condition (see Sec. 2.2)
6: Add the corresponding node to W if it fails.
7: Return W.

[1] http://www.nrel.gov/continuum/analysis/ergis.html

2.3 Problem Definitions

As mentioned in the introduction, identifying critical nodes that can cause maximum damage is important for improving the system reliability and efficiency. Hence given the cascade model, we formally define the following failure maximization problems to find such critical nodes.

Problem 1 (Max-Sub)

Given the heterogeneous network G, the failure cascade model F-Cas, and k.

Find the best set S^* of k transmission nodes to fail, s.t the expected number of final failed substations are maximized, i.e.

$$S^* = \arg\max_S \mathbb{E}[\#s|S] \tag{3}$$

where $\#s$ represents the number of substations that would eventually fail given the initial failure of S. Note that in **Max-Sub**, we select nodes based on the failure of substations because the loss/failure of an electrical substation directly leads to the power blackout of a region. In **Max-SubBus** defined below, we further extend **Max-Sub** by adding another component into our target.

Problem 2 (Max-SubBus)

Given the heterogeneous network G, the failure cascade model F-Cas, and k.

Find the best set S^* of k transmission nodes to fail, s.t the expected number of final failed substations, *and the transmission bus nodes* are maximized, i.e.

$$S^* = \arg\max_S \mathbb{E}[\#s + \#t|S] \tag{4}$$

Similar as in **Max-Sub**, $\#t$ represents the number of transmission nodes that would eventually fail given the initial failure of S.

3 OUR METHODS

The challenges of solving **Max-Sub** and **Max-SubBus** are two-fold. First, as described before, the failure does not necessarily cascade locally from one node to its neighbor. For example, we need to check the entire transmission network to decide if a substation fails or not, which is a very expensive operation. Second, failures loop through different components before convergence, making it harder to analyze. In fact, we can show that a well-known NP-hard problem (Influence Maximization for IC [20]) is a special case of both **Max-Sub** and **Max-SubBus** (by constructing a substation and a self loop for each transmission node[2]). Therefore, our problems are much more general than the influence maximization problems, and they are also NP-hard.

Lemma 3.1. ***Max-Sub*** *and* ***Max-SubBus*** *are NP-hard.*

Hence, we first attempt to solve them in a simplified scenario where the failure cascade does not form a loop. In such a scenario, instead of using G, we use G' which is a subgraph of G that only contains three components: power plants, transmission bus nodes, and substations. When only considering these three components, the failure of a substation cannot further induce failure of power plants, i.e. the failure spreads in one direction without forming any loop. In the following, we solve **Max-Sub** and **Max-SubBus** under such a simplified scenario first, and then propose our algorithms for the original problems where the failure cascade forms a loop.

[2]Detailed proof and additional results in appendix: https://goo.gl/QiTJMQ

3.1 Max-Sub and Max-SubBus without loop

In this section, we consider G' with only three CI components, and the failure cascades in one direction from the transmission network to the substations.

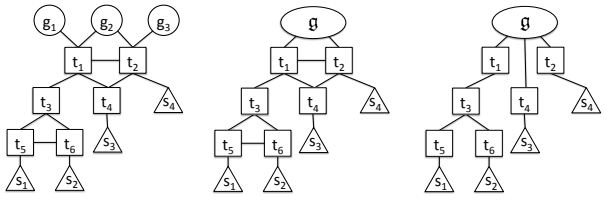
In both **Max-Sub** and **Max-SubBus**, we need to optimize on the expected number of failed substations, which can be written as

$$\mathbb{E}[\#s|S] = \sum_{s_i} \Pr(s_i|S) \tag{5}$$

where $\Pr(s_i|S)$ represents the probability of s_i's failure given the node set S which initially fail. By summing the failure probabilities over all substations, we get the expected number of failed substations. The failure probability $\Pr(s_i|S)$ basically represents the probability of s_i does not have a path to any power plants. We may combine connected component analysis and Trans-real, Trans-naive simulations to empirically estimate these probabilities, however it is hard to express them in a close form and hence hard to directly optimize. To express such a probability, we propose to use the dominator tree to capture critical nodes to s_i and estimate $\Pr(s_i|S)$.

As a first step, we merge all the power plants into a super power plant node $\mathfrak{g}$ (Fig. 2(a)(b)). The super node $\mathfrak{g}$ inherits all the edges from the merged power plants, i.e. if there is an edge connecting g_i to t_j, we create a link between $\mathfrak{g}$ and t_j. With such merging operation, the failure condition of a substation does not change: if a substation has a path to $\mathfrak{g}$, it certainly has a path to at least one of the original power plant; What's more, it allows us to construct a dominator tree rooted at $\mathfrak{g}$.

Dominator tree (D). In graph theory, given *any directed graph* and a starting node $\mathfrak{g}$, a node u dominates another node v if all paths from $\mathfrak{g}$ to v pass u. If all dominator nodes of v dominates u, then u is a direct dominator of v, denoted by $u = idom(v)$. For example, in Fig. 2(b), t_6 is a direct dominator of s_2 since all paths from $\mathfrak{g}$ to s_2 pass through node t_6, and all other dominators of s_2 dominates t_6. We can build a dominator tree rooted at $\mathfrak{g}$ by adding edges between all node pairs u and v if $u = idom(v)$. Dominator trees have been extensively studied in control-flow problems, and building dominator trees is a well-studied topic with near-linear time algorithms available [11].



(a) Original graph **(b) Get root $\mathfrak{g}$** **(c) Dominator tree**

Figure 2: Dominator tree examples. We first merge all power plants into the super root node $\mathfrak{g}$, and then construct the corresponding D.

Naturally, the substations would become leaf nodes in the constructed dominator tree. For each substation s_i there exists a path $P_i = \{\mathfrak{g}, t_1, t_2, ..., s_i\}$ in the dominator tree, which starts from $\mathfrak{g}$, goes over the transmission network and finally reaches the substation s_i. Further, each $t_j \in P_i$ dominates s_i, namely if any $t_j \in P_i$ fails, s_i would certainly fail. Therefore, we can estimate $\Pr(s_i|S)$ using its

ancestor nodes in the dominator tree as

$$\Pr(s_i|S) = 1 - \prod_{t_j \in P_i} (1 - \Pr(t_j|S)) \tag{6}$$

which is the probability of any transmission node t_j in P_i failing. Note that to simplify analysis, we assume independence among $\Pr(t_i|S)$. Since we know that if any $t_j \in P_i$ fails, s_i is bound to fail in the end, this is a lower bound estimation of the true $\Pr(s_i|S)$. Using this estimation, we can now reformulate our objective function (expected number of failed nodes) in **Max-Sub**, **Max-SubBus** (without loop) as

$$\mathbb{E}[\#s|S]' = \sum_{s_i} \Pr(s_i|S) = T - \sum_{s_i} \prod_{t_j \in P_i} (1 - \Pr(t_j|S)) \tag{7}$$

$$\begin{aligned}\mathbb{E}[\#s + \#t|S]' &= \sum_{s_i} \Pr(s_i|S) + \sum_{t_i} \Pr(t_i|S) \\ &= T - \sum_{s_i} \prod_{t_j \in P_i} (1 - \Pr(t_j|S)) + \sum_{t_i} \Pr(t_i|S)\end{aligned} \tag{8}$$

For both $\mathbb{E}[\#s|S]'$ and $\mathbb{E}[\#s + \#t|S]'$, we show that they satisfy the following diminishing return property[2].

Lemma 3.2. *$\mathbb{E}[\#s|S]'$ and $\mathbb{E}[\#s + \#t|S]'$ are both a) monotonically non-decreasing; b) sub-modular in terms of S.*

HotSpots framework. See pseudo-code of our HotSpots framework in Alg. 2. As both $\mathbb{E}[\#s|S]'$ and $\mathbb{E}[\#s + \#t|S]'$ are submodular, this immediately gives us a $(1 - 1/e)$-*approximation algorithm* for optimizing them [23]. HotSpots uses a greedy method to iteratively select nodes that lead to the maximum marginal gain of the final objective function.

To calculate the marginal gain $\delta(t_i)$ of adding a node t_i to S, we first need to estimate $\Pr(t|S)$ for calculating $\mathbb{E}[\#s]'$ and $\mathbb{E}[\#s + \#t]'$. For this purpose, we run a set of m Trans-real/Trans-naive simulations on the transmission network to get the empirical probabilities of a t_i failing given S. Then, we propose to use a recursive function to efficiently calculate the value of the objective function given $\Pr(t|S)$ (Alg. 4). In effect, this recursive function just does a correct 'walk' on the dominator tree. We observe that the objective function Eq. 7 (and also the first part of Eq. 8) can be naturally factorized over different paths in the dominator tree (each t corresponds to a path in D). Therefore, we call our $recur(\cdot)$ function on the root node $\mathfrak{g}$, and it goes over the dominator tree in a top-down fashion, where we take summation when iterating horizontally (visiting the children of a node), and production when iterating vertically (visiting different nodes in the same path). Combining the above, we calculate the marginal gain $\delta(t_i)$ of adding a node in line $15-16$. In practice, we also adopt a lazy evaluation strategy [21] to speed up the algorithm (by avoiding evaluating $\delta(t_i)$ for all t_i in each iteration). For ease of understanding, we omit it in Alg. 2.

Remark. The original **Max-Sub** and **Max-SubBus** do not satisfy these properties. One of our main contributions is to use the dominator-tree-based method to reformulate and estimate the optimization problems s.t. they satisfy the diminishing return property, and thus can be solved near-optimally using a greedy algorithm.

3.2 Max-Sub and Max-SubBus with loop

Now we consider the original G with all five components, and the failures can further spread from substations to natural gas compressors, power plants, and finally to substations again (the 'loop'). Our main idea is to first calculate the failure probabilities

Algorithm 2 HotSpots framework

Input: G', F-Cas, k, m
Output: A set S of k nodes
1: $S = \{\}$
2: Merge all power plants into $\mathfrak{g}$, and construct a dominator tree D rooted at $\mathfrak{g}$ for G'
3: **while** $|S| < k$ **do**
4: **for** each $t_i \notin S$ **do**
5: Estimate $\Pr(t|S)$, $\Pr(t|S \cup t_i)$ using F-Cas simulation.
6: $\delta(t_i) = Update(\Pr(t|S \cup t_i), \Pr(t|S), D, \mathfrak{g})$
7: $t^* = \arg\max \delta(t_i)$, add t^* to S
8: Return S

Algorithm 3 $Update(\cdot)$

Input: $\Pr(t|S \cup t^*)$, $\Pr(t|S)$, D/D^+, $\mathfrak{g}$
Output: $\delta(t^*)$
//**For the without-loop version**
1: Return $Recur(\Pr(t|S \cup t^*), D, \mathfrak{g}) - Recur(\Pr(t|S), D, \mathfrak{g})$
//**For the with-loop version**
2: Initialize all $\Pr(s|S \cup t^*)$, $\Pr(s|S)$, $\Pr(d|S \cup t^*)$, $\Pr(d|S)$ as 0
3: **while** $\Pr(s|S \cup t^*)$, $\Pr(s|S)$ is changing **do**
//Traverse D^+ to update $\Pr(s|S \cup t^*)$, $\Pr(s|S)$
4: $Recur^+(\Pr(s|S \cup t^*), \Pr(t|S \cup t^*), D^+, \mathfrak{g}, 1)$
5: $Recur^+(\Pr(s|S), \Pr(t|S), D^+, \mathfrak{g}, 1)$
6: Update $\Pr(d|S \cup t^*)$, $\Pr(d|S)$ using Eq. 10
7: Return $\mathbb{E}[\#s|S \cup t^*] - \mathbb{E}[\#s|S]$ using Eq. 11 (similarly for **Max-SubBus**)

Algorithm 4 $Recur(\Pr(t|S), D, x)$

Input: $\Pr(t|S)$, the dominator tree D, the current node visited x
Output: The value of the objective function Eq. 7 (Eq. 8 can be calculated similarly)
1: **if** x is a substation **then**
2: Return 1
3: $kids$ = {child nodes of x in D }
4: **if** $kids = \emptyset$ **then**
5: Return 0//reach a leaf node in D that is not a substation
6: $s = 0$//add up the value from the child nodes
7: **for** each kid in kids **do**
8: $s+ = Recur(\Pr(t|S), D, kid)$//call Recur() on the child node
9: Return $\Pr(x|S) * s$

of power plants given those of substations using the natural gas system. Then we modify the dominator tree we constructed above to include the failure of power plants, and thus encode the failure cascade from substation to power plants.

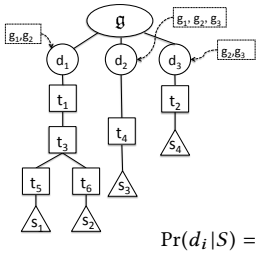
3.2.1 Failure probabilities of power plants. In the failure cascade loop, once a substation fails, the natural gas compressors that rely on it for power would fail. Similarly, once a natural gas compressor fails, the power plants it supplies fuel to (through the pipeline network) may fail. As mentioned in Sec. 2.2, we assume the pipelines do not fail; hence we can directly map each power plant g_i to a set of relevant substations, and derive its failure probability.

$$\Pr(g_i|S) = 1 - \prod_{c_j \in A_i} (1 - \Pr(Sub(c_j)|S)) \tag{9}$$

where A_i is the set of compressor nodes that fuel the power plant g_i through the pipeline network, and $Sub(c_j)$ represents the substation

that is connected to c_j. Basically, for each power plant, we find all the natural gas compressors it can reach through the pipeline network, and find the substations that are connected with these natural gas compressors, and finally calculate the probabilities accordingly.

3.2.2 Dominator tree with loop (D^+). The dominator tree we had constructed before merges all power plants to a super root node g. While this is an essential step to construct a meaningful dominator tree for our tasks, it brings difficulties to include failure probabilities of power plants since all of them are represented as g. To modify the dominator tree to include the failure probabilities of power plants, we insert a dummy node for each branch in D to represent the set of power plants that provide power to this branch. For example in the left snippet, for each direct child node t_i of g in D, we insert a node d_i, such that the connection from g to t_i would go through d_i first. This dummy node d_i represents the set of power plants that can reach t_i, and it fails with the following probabilities:



$$\Pr(d_i|S) = 1 - \prod_{g_j \in B_i} (1 - \Pr(g_j|S)) \tag{10}$$

where B_i represents the set of power plants that can reach t_i.

With the addition of these dummy nodes, we now have a failure cascade loop in the dominator tree. Initially $\Pr(d|S)$ are set as 0; then we estimate the failure probabilities of substations $\Pr(s|S)$; then we update the value of $\Pr(d|S)$, and recalculate $\Pr(s|S)$, so on. These probabilities are non-decreasing over iterations in the failure cascade loop, and they are bounded by 1. Hence we would eventually converge. We can rewrite our objective function Eq. 7, Eq. 8 using the probabilities $\hat{\Pr}(s|S)$ after convergence.

$$\mathbb{E}[\#s|S]^+ = T - \sum_{s_i} \prod_{t_j \in P_i} (1 - \Pr(t_j|S)) \cdot \prod_{g_j \in B_i} \prod_{c_y \in A_j} (1 - \hat{\Pr}(Sub(c_y)|S)) \tag{11}$$

$$\mathbb{E}[\#s + \#t|S]^+ = T - \sum_{s_i} \prod_{t_j \in P_i} (1 - \Pr(t_j|S)) \cdot \prod_{g_j \in B_i} \prod_{c_y \in A_j} (1 - \hat{\Pr}(Sub(c_y)|S)) + \sum_{t_i} \Pr(t_i|S) \tag{12}$$

Further, we can prove by induction that the above objective functions maintain the diminishing return property.

Lemma 3.3. *$\mathbb{E}[\#s|S]^+$ and $\mathbb{E}[\#s + \#t|S]^+$ are both a) monotonically non-decreasing; b) sub-modular in terms of S^2.*

Hence, the complete problem with loop can be solved using the same $(1 - 1/e)$-approximation framework in Alg. 2, with a new dominator tree, $Update(\cdot)$, and $Recur^+(\cdot)$ function (Alg. 3, Alg. 5).

The overall time complexity of our HotSpots framework is $O(kV_t(mE_t + lV_t + lV_s))$, where V_t, V_s are the number of transmission nodes and substation nodes respectively, E_t is the number of edges in the transmission network, and l is the number of failure cascade loops until the probabilities converge.

Algorithm 5 $Recur^+(\Pr(s|S), \Pr(t|S), D^+, x, v)$

Input: $\Pr(s|S), \Pr(t|S), D^+, x, v$ (the probability that none of the previous nodes fail)
Output: Update the values of $\Pr(s|S)$

```
if x is a substation then
    Pr(x|S) = 1 − v
else
    kids = {child nodes of x in D+ }
    for each kid in kids do
        Recur+(Pr(s|S), Pr(t|S), D+, kid, v ∗ (1 − Pr(x|S)))
```

Node Type	TN	PA	FL	OH
Power Plants	11	45	79	35
Transmission Nodes	206	224	253	105
Electrical Substations	489	831	1590	806
Gas Compressors	105	291	45	189
Pipelines	387	5667	624	7641

Table 2: Number of nodes in each CI components for each dataset we use.

4 EXPERIMENTS

In this section, we design various experiments and case studies to evaluate our algorithms. For the experiments, we set the load and the capacity of any transmission node as a constant value. Different load and capacity settings can easily be used by changing the corresponding values in F-Cas.

Datasets: We construct heterogeneous CI networks as described in Sec. 2.1 for four different states for our evaluation: Tennessee (TN), Pennsylvania (PA), Florida (FL) and Ohio (OH). The statistics of each of these datasets are shown in Tab. 2.

Baselines: To the best of our knowledge, there is no existing algorithm that can be used to solve the failure maximization problems (**Max-Sub**, **Max-SubBus**) as our algorithms do for the CI networks. We adapt two related algorithms, and generate several algorithms with different node selection strategies as our baselines.

(1) Opera [13] is a recent work which picks k critical nodes that would maximally break the connectivity of a target network (which can be measured by the number of triangles in the network). We run it on the transmission network.

(2) Netshield [27] is an immunization algorithm which aims to minimize the epidemic threshold of the graph. We select the top k transmission nodes according to the ranking given by the algorithm.

(3) Degree: pick transmission nodes with the highest degrees.

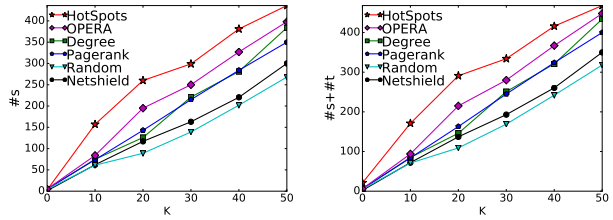
(4) Pagerank: pick transmission nodes with the highest pageranks.

(5) Random: We randomly select k transmission nodes.

4.1 Effectiveness (Q1)

For the nodes selected by our algorithms and the baselines, we run our F-Cas simulation to evaluate the effectiveness for **Max-Sub** and **Max-SubBus**. See Figure 3 (only show OH for lack of space, results for FL, TN, PA are similar[2]). In all states under all situations (Trans-real and Trans-naive), the proposed HotSpots algorithm performs better than the baseline methods. The difference between HotSpots and other baselines decreases as k increases. This is expected, as more seeds are picked, the eventual failures will saturate. Note that Opera does not give good results for **Max-Sub** and **Max-SubBus**, as it is designed to deal with static interdependent networks. Our

Max-Sub and **Max-SubBus** both need the dynamics of the whole network system—as a result, our proposed HotSpots outperforms Opera. Also for other baselines, as they only consider the static information within one network, they do not perform as well or as stable as HotSpots.



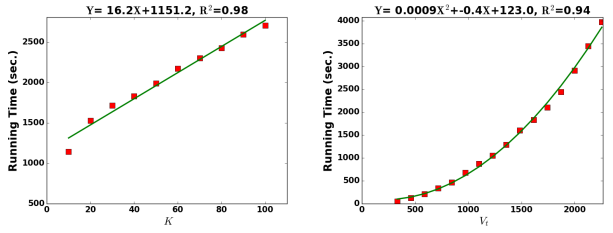
(a) OH, Max-Sub (b) OH, Max-SubBus

(c) OH, Max-Sub (d) OH, Max-SubBus

Figure 3: Evaluating the detected nodes. Proposed HotSpots (in red), outperforms all the baselines. Top/bottom row are Trans-naive/Trans-real. # of seeds k vs expected # of substations (and transmission nodes) from each method: larger is better.

4.2 Scalability (Q2)

We investigate how HotSpots scales as the number of seeds k and as size of network $|V|$ changes (we ran HotSpots on increasing sizes of the TN state data). See Figure 4. As expected from our complexity analysis, it scales linearly with k and almost quadratically on $|V|$. Note that for all our datasets, HotSpots finished within 30 minutes in choosing top-50 nodes.



(a) vs k (b) vs $|V|$

Figure 4: Scalability Results.

4.3 Success Stories and Case Studies (Q3)

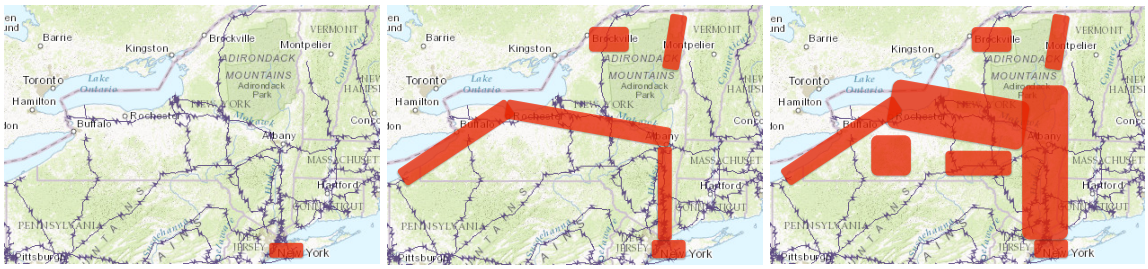
In this section, we show how HotSpots algorithms and results can be used for real applications at ORNL and beyond. Note that the HSIP Gold data we used is not public, and we cannot directly show the network we constructed due to security reasons. Therefore in all our visualizations, we use the maps and the high-voltage transmission lines from the US Energy Information Administration (EIA) [3] which are publicly available.

Heterogeneous networks for CI. Constructing a heterogeneous CI network having interlinks within and across networks to capture various physical, geographical and cyber interdependencies provides useful insights right off the bat. For example, by simply overlaying the hurricane advisory track over the heterogeneous network we constructed, we can quickly estimate the extent of the infrastructure damage, and even predict the spread of the damage beyond the contour of the hurricane track. This is crucial as getting such estimates currently are non-trivial, and time consuming [10].

In this case study, we overlay actual hurricane sandy wind swath tracks (advisory no. 20) obtained from Hurricane Mapping [4] over our five-component heterogeneous network, and estimate both the immediate and predicted damage of the hurricane. The results from our analysis provide several intuitive insights. Firstly, when the hurricane Sandy makes landfalls on New York city, by simply overlaying the hurricane track with our G, we can quickly estimate the CI facilities directly affected by the hurricane. As shown in Fig. 5(a), we expect from HotSpots that 227 nodes in G are affected after sandy hits NY city. Assuming these 227 nodes would fail immediately, by running our failure cascade model F-Cas, we can predict the future damage of the CI network and pinpoint high risk regions and facilities. In Fig. 5(b)(c), the failure quickly cascades to other CI facilities in the entire NY state. Further, the cascading and escalating loop failures are clearly visible on the map with 487 nodes failing before the failure cascade loop is formed and 712 nodes failing after the loop formation. These reinforcing loops were one of the major causes for the prolonged recovery. Our failure model is able to systematically predict the cascading loop trends given the initial perturbation input. Such prediction results complement existing hurricane assessment tools, such as HEADOUT [2], OCIA[3], EARRS [8] by including the failure cascade affect into their damage assessment system which are mainly based on application of fragility curves [7] without considering the interdependencies among different CI components.

Comparison with the 2003 Blackout. An in-depth study on the US NE 2003 blackout event revealed that a single high voltage transmission line in northern Ohio brushed against some overgrown trees and shut down due to overheating [5]. Within a couple of hours, three other lines sagged into trees and switched off, forcing other power lines to shoulder an extra burden and tripping a cascade of failures throughout southeastern Canada and eight NE states. This suggests that the initiators of this massive blackout should be critical nodes in the context of cascading failures. In this case study, we select a portion of the heterogeneous network overlapping the Ohio region and run HotSpots to identify the top 5 vulnerable nodes (shown in Fig. 6a). By comparing our findings with the study mentioned above, we find that one of the nodes (on the top right) we identified are truly critical: it is only one hop away (within ~ 10 miles in geographical distance) from the nodes connecting the 3 transmission lines which actually failed and triggered the blackout. Such critical nodes identified by HotSpots can also support DHS NIPP[4] for strengthening the security and resilience of the CI system. In contrast, the baselines do not detect these critical nodes. For

[3] https://www.dhs.gov/office-cyber-infrastructure-analysis
[4] https://www.dhs.gov/sites/default/files/publications/NIPP%202013_Partnering%20for%20Critical%20Infrastructure%20Security%20and%20Resilience_508_0.pdf

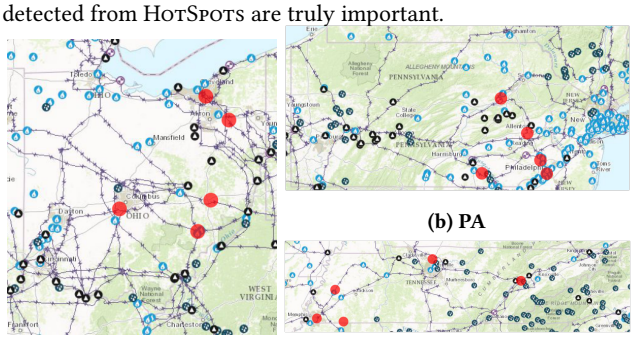


(a) 227 **nodes initially affected in NY city (b)** 487 **nodes fail before the cascade loop (c)** 712 **nodes fail after a loop of cascade**

Figure 5: Estimate the impact of hurricane sandy on NY state. (a) # of directly affected nodes by overlaying constructed G with the hurricane track (b) F-Cas simulation results before the failure cascade forms a loop. (c) F-Cas results after the failure cascade forms a loop. We mark regions with high number of CI failures as red (actual network not shown for data privacy).

example, Netshield detects much different nodes that are more than 80 miles away from the critical locations in the blackout.

Analyzing detected critical nodes. To evaluate the quality of the critical nodes HotSpots finds, we mark our results on high-voltage (> 345 kV) transmission networks sourced from the EIA. In Fig. 6, the locations of the detected nodes are marked with a red circle. While the high-voltage transmission lines are not a direct indicator of importance, it is expected in general that critical nodes would be either on large generation plants or on transmission line terminations with a large power throughput. In the visualization of the high-voltage transmission networks, this is often correspondent to nodes with several *converging lines*. In these figures, we clearly observe that many of the top transmission nodes HotSpots identified are on these high-voltage transmission lines, and some on the intersection of multiple HV lines, indicating that the nodes detected from HotSpots are truly important.



(a) OH **(b) PA** **(c) TN**

Figure 6: Top five transmission nodes identified by HotSpots, highlighted with red circles.

4.4 User interface

To improve usability, we also spent time designing a web-based user interface that integrates all the proposed tools and algorithms and also facilitates 'what-if' scenario analysis. We describe the details in the following and show an example screenshot in Fig. 7.

Our interface allows users to input their own CI data, and construct their own heterogeneous CI network using our graph generation toolkit (urbannet-toolkit). The heterogeneous network will then be visualized on a real map (top right in Fig. 7), with options to show a certain type of CI component (top left), and to check detailed attributes of a specific node. Further, it allows users to input a perturbation (i.e. selecting initial failure nodes), customize and run our failure cascading simulation (bottom left), and get real-time failure statistics and visualizations in all the CI components (bottom right). Finally, it integrates the HotSpots algorithm which automatically identifies and visualizes the critical nodes. We also provide a knob which can control the number of critical nodes the analyst wants to visualize. This interface would greatly help in analyzing the vulnerability of the CI systems for domain experts and decision makers.

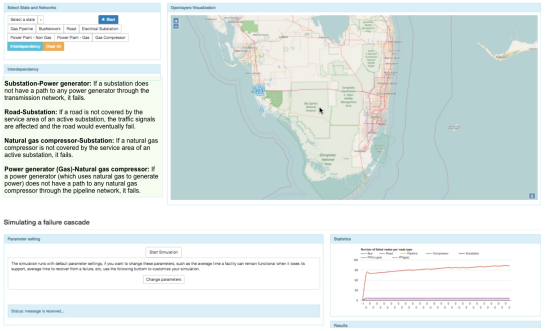


Figure 7: Snapshot of our UI (shows part of Florida).

5 DISCUSSION

To the best of our knowledge, the proposed HotSpots algorithm, as well as our urbannet-toolkit and F-Cas model are the first attempt in analyzing up to five CI components in a unified framework from a graph analytic view point. Additional CI components can be easily added given the interlinking rules. The graphs so generated facilitate network-based analysis and easy visualization for better understanding. As discussed in Section 2.2, our failure cascade model F-Cas is not very expensive to run, and captures both the path-based and neighbor-based failure conditions, that cannot be modeled by existing cascade models. Note that such path-based failure cascading is not restricted to the transmission network. Similar overload phenomenons and path-based failures can be observed in the electrical distribution network (crucial for power delivery), which distributes power from substations to local facilities (a layer that we do not model to simplify our analysis). Take the SCADA system (supervisory control and data acquisition)[5] as another example. The communication servers send control signals through the WAN and LAN network to control facilities such as PLCs (programmable logic controllers) and RTUs (remote terminal units). When some

[5]https://en.wikipedia.org/wiki/SCADA

connections in the WAN and LAN network are damaged, message flooding may happen which would cause message loss (similar to the overloading in our problem), and further cause malfunctioning in the electric grid system. As a result, HotSpots can be easily applied on a wide range of CI systems to identify the critical nodes in those systems.

6 OTHER RELATED WORK

In addition to work that we have mentioned in previous sections, we discuss more related work here.

Infrastructure Vulnerability Analysis. Previous works on vulnerability analysis and simulation of interdependencies between critical infrastructure systems can be categorized into: empirical approach, agent based, system dynamics based, economic theory based, and network based approaches (see [24] for a review). A few mathematical frameworks [12] and interdependency models [17] have been proposed for vulnerability analysis. Most of these work focus on only two critical infrastructures at a time. For example, Parandehgheibi et al. [25] study the interdependency between a communication network and a power grid network, Dueñas-Osorio et al. [18] study the fragilities and interdependency between power and water network. We study a more general problem where we focus on combining multiple practical (five) CI components together.

Influence Maximization and Cascade Analysis. The influence maximization problem aims to find the best seed nodes which maximize influence. This has been extensively studied on the Independent Cascade and the Linear Threshold models [20], where they gave a (1-1/e) approximation algorithm based on submodularity. Much work has focused on designing more efficient algorithms for the original problem [9], or extending to continuous time models [16], or under uncertainty [15]. All these algorithms assume that the influence cascades locally through edges. In contrast, in our problem, the failure condition for nodes requires examination of the connectivity of the entire network, and not just neighbors. A very recent work Opera [13] finds critical nodes in a CI network that would maximally decrease the connectivity in target networks. However, they still use a 'local' failure model, which does not capture the dynamics of CIS (they optimize on a different objective function based on the number of triangles in the network).

7 CONCLUSIONS

In this paper, we approach the problem of failure analysis on CI systems using heterogeneous networks. We construct real CI networks from datasets at ORNL, and formulate a novel cascade model F-Cas and problems to identify critical nodes. We then develop an effective and scalable algorithm HotSpots for these problems. We also showed through extensive experiments on real datasets, that such an approach is useful for CI analysis, that HotSpots gives high quality results beating competitors, and the results are interpretable and meaningful matching real-world situations.

As future work, the results from our approaches can serve as an input starting point for expensive, high-fidelity simulations. Further, applying HotSpots to other CIs such as the transportation network and the SCADA system is also useful.

Acknowledgments. This paper is based on work partially supported by the NSF (IIS-1353346), the NEH (HG-229283-15), ORNL (Order 4000143330) and from the Maryland Procurement Office (H98230-14-C-0127), and a Facebook faculty gift.

REFERENCES

[1] Homeland security infrastructure program (HSIP). https://gii.dhs.gov/HIFLD/hsip-guest.
[2] Hurricane electrical assessment damage outage tool (HEADOUT). http://www.gss.anl.gov/wp-content/uploads/2015/09/MORS_Presentation_Talaber_WG21_and_WG31_060415.pdf.
[3] U.S. energy information administration (EIA). https://www.eia.gov/.
[4] U.S. hurricane mapping. https://hurricanemapping.com/.
[5] S. Abraham, H. Dhaliwal, R. J. Efford, L. J. Keen, A. McLellan, J. Manley, K. Vollman, N. J. Diaz, T. Ridge, et al. *Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations*. US-Canada Power System Outage Task Force, 2004.
[6] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the north american power grid. *Phys. Rev. E*, 69:025103, feb 2004.
[7] M. Allen, S. Fernandez, O. Omitaomu, and K. Walker. Application of hybrid geo-spatially granular fragility curves to improve power outage predictions. *J Geogr Nat Disast*, 4(127):2167–0587, 2014.
[8] A. M. Barker, E. B. Freer, O. A. Omitaomu, S. J. Fernandez, S. Chinthavali, and J. B. Kodysh. Automating natural disaster impact analysis: An open resource to visually estimate a hurricane's impact on the electric grid. In *Southeastcon*, pages 1–3, 2013.
[9] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, pages 946–957, Philadelphia, PA, USA, 2014.
[10] W. N. Bryan. *Hurricane Sandy Situation Report*. U.S. Department of Energy Office of Electricity Delivery & Energy Reliability, 2012.
[11] A. L. Buchsbaum, H. Kaplan, A. Rogers, and J. R. Westbrook. A new, simpler linear-time dominators algorithm. *ACM Trans. Program. Lang. Syst.*, 20(6):1265–1296, 1998.
[12] S. V. Buldyrev, N. W. Shere, and G. A. Cwilich. Interdependent networks with identical degree of mutually dependent nodes. *Physical Review*, 83(1), 2011.
[13] C. Chen, J. He, N. Bliss, and H. Tong. On the connectivity of multi-layered networks: Models, measures and optimal control. In *ICDM*. IEEE, 2015.
[14] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He. Fascinate: Fast cross-layer dependency inference on multi-layered networks. In *KDD*. ACM, 2016.
[15] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou. Robust influence maximization. In *KDD*. ACM, 2016.
[16] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schoelkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML*, pages 793–801, 2014.
[17] S. Duan, S. Lee, S. Chinthavali, and M. Shankar. Reliable communication models in interdependent critical infrastructure networks. In *Resilience Week (RWS), 2016*, pages 152–157. IEEE, 2016.
[18] L. DueÃśas-Osorio, J. I. Craig, and B. J. Goodno. Seismic response of critical interdependent networks. *Earthquake Engineering and Structural Dynamics*, 36(2):285–306, 2007.
[19] A. Dwivedi and X. Yu. A maximum-flow-based complex network approach for power system vulnerability analysis. *IEEE Transactions on Industrial Informatics*, 9(1):81–88, 2013.
[20] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*. ACM, 2003.
[21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*. ACM, 2007.
[22] Y.-S. Li, D.-Z. Ma, H.-G. Zhang, and Q.-Y. Sun. Critical nodes identification of power systems based on controllability of complex networks. *Applied Sciences*, 5(3):622–636, 2015.
[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions i. *Mathematical Programming*, 14(1):265–294, 1978.
[24] M. Ouyang. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering and System Safety*, 121:43–60, 2014.
[25] M. Parandehgheibi and E. Modiano. Robustness of bidirectional interdependent networks: Analysis and design. *arXiv preprint arXiv:1605.01262*, 2016.
[26] A. Sen, A. Mazumder, J. Banerjee, A. Das, and R. Compton. Identification of k most vulnerable nodes in multi-layered network using a new model of interdependency. In *INFOCOM WKSHPS*, pages 831–836. IEEE, 2014.
[27] H. Tong, B. A. Prakash, C. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau. On the vulnerability of large graphs. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1091–1096. IEEE, 2010.
[28] U.S. Department of Energy. The Water-Energy Nexus: Challenges and Opportunities, June 2014.
[29] Y. Yuan, Z. Li, and K. Ren. Modeling load redistribution attacks in power systems. *IEEE Transactions on Smart Grid*, 2(2):382–390, 2011.