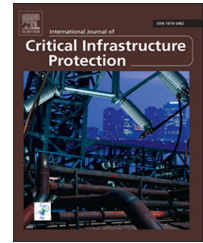


Available online at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/ijcip

Time-based critical infrastructure dependency analysis for large-scale and cross-sectoral failures



George Stergiopoulos^a, Panayiotis Kotzanikolaou^b, Marianthi Theocharidou^{c,*},
Georgia Lykou^a, Dimitris Gritzalis^a

^aInformation Security and Critical Infrastructure Protection Laboratory, Department of Informatics, Athens University of Economics and Business, 76 Patission Avenue, GR-10434 Athens, Greece

^bDepartment of Informatics, University of Piraeus, 85 Karaoli and Dimitriou, GR-18534 Piraeus, Greece

^cSecurity Technology Assessment Unit, Institute for the Protection and the Security of the Citizen, European Commission Joint Research Centre, via E. Fermi 2749, I-21027 Ispra, Italy

ARTICLE INFO

Article history:

Received 30 January 2015

Received in revised form

8 December 2015

Accepted 8 December 2015

Available online 14 December 2015

Keywords:

Critical infrastructures

Cascading failures

Dependency risk graphs

Time analysis

Resilience

Fuzzy logic

ABSTRACT

Dependency analysis of critical infrastructures is a computationally intensive problem when dealing with large-scale, cross-sectoral, cascading and common-cause failures. The problem intensifies when attempting a dynamic, time-based dependency analysis. This paper extends a previous graph-based risk analysis methodology to dynamically assess the evolution of cascading failures over time. Various growth models are employed to capture slow, linear and rapidly evolving effects, but instead of using static projections, the evolution of each dependency is “objectified” by a fuzzy system that also considers the effects of nearby dependencies. To achieve this, the impact (and, eventually, risk) of each dependency is quantified on the time axis into a form of many-valued logic. In addition, the methodology is extended to analyze major failures triggered by concurrent common-cause cascading events. A critical infrastructure dependency analysis tool, CIDA, that implements the extended risk-based methodology is described. CIDA is designed to assist decision makers in proactively analyzing dynamic and complex dependency risk paths in two ways: (i) identifying potentially underestimated low risk dependencies and reclassifying them to a higher risk category before they are realized; and (ii) simulating the effectiveness of alternative mitigation controls with different reaction times. Thus, the CIDA tool can be used to evaluate alternative defense strategies for complex, large-scale and multi-sectoral dependency scenarios and to assess their resilience in a cost-effective manner.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Most critical infrastructures (CIs) can be modeled as cyber-physical systems whose cyber components control their

underlying physical components. Critical infrastructures are inherently complex systems because they integrate heterogeneous platforms, proprietary systems and protocols, and open communications networks. In addition, critical

*Corresponding author.

E-mail address: marianthi.theocharidou@jrc.ec.europa.eu (M. Theocharidou).

infrastructures are usually interconnected and interdependent with other critical infrastructures that belong to other sectors (e.g., energy, information and communications technology (ICT) and transportation). According to Rinaldi et al. [33], critical infrastructures may have physical, informational and logical dependencies. Specifically, a failure in one infrastructure may affect the operation of other critical infrastructures as a result of their dependencies. Note that a failure is meant in a broad sense and covers an accidental failure, natural disaster or deliberate cyber attack [22]. In the case of a geographical dependency, seemingly independent critical infrastructures may be affected by a threat due to their physical proximity. Protecting against such types of failures is an active area of research as manifested by the numerous projects on the topic, including DIESIS [35,47], I2Sim [27] and CIPRNet [8]. Dependency modeling, simulation and analysis have been studied extensively by researchers. A recent publication [30] surveys current approaches and classifies them into several broad categories, including empirical, agent based, system dynamics based, economic theory based and network based approaches, among others.

Disruptions or outages in critical infrastructures are usually categorized as cascading, escalating or common-cause failures [33]. A cascading failure occurs when an infrastructure A affects one or more components in another infrastructure B, which, in turn, leads to the partial or total unavailability of infrastructure B. An escalating failure occurs when a disruption in one infrastructure exacerbates an independent disruption in another infrastructure, usually by increasing the severity of the disruption and/or the time needed to recover from the second failure. A common-cause failure occurs when two or more infrastructures are disrupted at the same time; components within each infrastructure fail due to a common cause. This occurs when two infrastructures are co-located (geographic interdependency) or when the root cause of the failure is widespread (e.g., a natural or human-initiated disaster).

1.1. Large-scale and cross-sectoral dependencies

The most common examples of large-scale failures caused by critical infrastructure dependencies involve power transmission networks, such as the major blackouts of 2003 in the United States, Canada and Europe [1]. The cascading process of a failure has been studied and modeled by several researchers [3,10,32,51,54,55]. Although there is a lack of statistical data for such failures, recent efforts (see, e.g., [49]) have produced failure statistics based on empirical data reported by the media. One of the key findings is that large-scale critical infrastructure cascading dependencies occur more frequently than expected. However, the effects do not often cascade deeply; specifically, critical infrastructure nodes that are four or five hops away in a dependency chain are rarely affected. Another key finding is that, although most reported initiators of cascading effects are critical infrastructure assets belonging to the energy and information and communications technology sectors, the cascading effects are primarily cross-sectoral in nature (i.e., critical infrastructure assets in multiple sectors are affected). This is reasonable because the source infrastructures (energy and

information and communications technology) usually provide vital services to other critical infrastructures in various sectors, thus creating multiple direct (or first-order) dependencies.

1.2. Motivation

In recent years, several dependency analysis methodologies and tools have been developed; these focus on the impact [13], consequences [34] or risk derived from critical infrastructure dependencies [18–21,48] and their potential cascading effects. The methodologies and tools are usually sector-specific and oriented towards power distribution (e.g., [5]) or water distribution networks (e.g., [17]). These tools are very useful for low-level analyses of small-scale scenarios (e.g., identifying the critical components within a power transmission network). However, they may fall short when high-level analyses are needed in order to model large-scale, cross-sectoral scenarios. One example is the identification of dependency paths of high economic or societal risk that affect multiple sectors.

Critical infrastructure asset owners and operators typically perform risk assessments at the organization level and may not have knowledge about (or interest in) threats that emanate from dependent critical infrastructures. This knowledge can be acquired, to some extent, by conducting international table-top exercises [26]. Nonetheless, although the economic impact of a failure can be assessed by a critical infrastructure owner or operator (organization-wise), the overall impact (or risk) of a given critical infrastructure failure on a dependent critical infrastructure is not a tangible value, especially when multi-order dependencies are present.

The need for high-level, multi-sectoral risk assessments has been recognized by several international bodies [12]. A high-level risk analysis allows the identification of complex cascade or common-cause risk paths and the comparison of alternative mitigation strategies. Note that a multi-layer risk assessment is not an alternative to organization-wide risk assessments because they are prerequisite inputs to a multi-layer risk analysis. Such an analysis requires the modeling and analysis of hundreds or even thousands of critical infrastructures. The complexity is even greater if a time-based analysis is to be performed. Unfortunately, the computation of the cumulative security risks and the identification of the critical points of failure are NP-complete and, thus, suboptimal (albeit useful) dependency analysis tools have to be developed.

1.3. Contributions

This paper extends recent work on critical infrastructure dependency analysis [19–21]. The first extension is the use of time-based analysis models to study the evolution of dependency chains during slow, linear and rapidly evolving cascading failures. Note that each dependency may follow a different time model that is “fine-tuned” to each examined dependency using fuzzy modeling. The second extension is the modeling of concurrent cascading and common-cause failures in order to effectively analyze major failures.

This paper also describes CIDA [42], a critical infrastructure dependency analysis tool that is based on risk analysis and graph modeling. In particular, CIDA is a proactive modeling and security dependency analysis tool for evaluating large-scale, cross-sectoral dependency scenarios. It allows risk assessors and decision makers to analyze complex dependency graphs and to identify critical dependency chains before an actual threat has occurred. Thus, it can reveal underestimated dependency risks that need further attention. CIDA may also be used to efficiently assess alternative risk mitigation strategies and, thus, enhance critical infrastructure resilience. In this work, resilience implies the ability to withstand accidental or deliberate threats or incidents [46].

In order to validate the applicability and efficiency of the extended critical infrastructure dependency analysis approach, the CIDA tool is stress-tested using random graphs of up to one thousand critical infrastructure nodes with randomly selected dependencies. The tests demonstrate the computational efficiency of CIDA in large-scale scenarios under reasonable parameters (maximum number of dependencies per node and maximum order of dependencies). As a proof of concept, targeted tests based on data from real cascading and common-cause failures are also conducted. Since a resilience-oriented approach acknowledges that failures will occur, critical infrastructures should implement controls that effectively absorb, adapt to and rapidly recover from disruptive events [14].

2. Building blocks

This section briefly describes the two main building blocks used in the proposed methodology: (i) the underlying multi-risk dependency analysis methodology for cascading failures; and (ii) the fuzzy modeling approach used for the time-based analysis of dependencies.

2.1. Multi-risk dependency analysis methodology

The multi-risk dependency analysis method [19–21] that forms the foundation of this work leverages the combined results of organization-level risk assessments performed by critical infrastructure owners and operators to assess the risk of n th-order dependencies. Directed graphs are used to

visualize the relationships (dependencies) between critical infrastructures.

2.1.1. First-order dependency risk

A dependency can be defined as a “one-directional reliance of an asset, system, network or collection thereof – within or across sectors – on an input, interaction or other requirement from other sources in order to function properly” [9]. In this work, dependencies are modeled using a graph $G=(N,E)$ where N is a set of nodes (infrastructures or components) and E is a set of edges (or dependencies). The graph is directional in nature to model dependencies from one critical infrastructure to other critical infrastructures. An edge from node CI_i to node CI_j , i.e., $CI_i \rightarrow CI_j$, denotes a risk relation that is derived from the dependence of infrastructure CI_j on a service provided by infrastructure CI_i . This relation is quantified using the impact I_{ij} and the likelihood L_{ij} of a disruption being realized. The product of these two values is defined as the dependency risk R_{ij} to infrastructure CI_j due to its dependence on infrastructure CI_i . The numerical value associated with each edge refers to the level of the cascade-resulting risk for the receiver due to the dependency. This risk is depicted using a risk scale [1..9] where 9 is the most severe risk. All the parameters (L_{ij} , I_{ij} , R_{ij}) are defined in order to assess the risk of first-order dependencies. The main input to this method is provided by critical infrastructure owners and operators, and refers to the obvious upstream dependencies as mentioned above.

The method for modeling multiple first-order dependencies is clarified using an example involving a mini-telecommunications blackout in Rome in 2004 [23]. The cause was a flood at a major telecommunications service node (Laurentina-Inviolatella) due to a broken pipe that provided water to the cooling plant. This caused several circuits to fail, including the main power supply. Backup power generators failed to start due to the presence of water and the batteries that provided power to electronic equipment were also damaged. The cooling plant had to be shut down to perform repairs, but this led to the overheating of several telecommunication devices.

The disruption of Laurentina-Inviolatella (node A) caused problems and delays in several infrastructures as shown in Table 1 (based on the description in [23]). These included Fiumicino Airport (node B), ANSI Print Agency (node C), several post offices (node D), banks (node E), the ACEA power distribution system (node F) and the communications

Table 1 – Mini telecommunications blackout – first-order dependencies.

Node: CI	Sector	First-Order Effects
A: Laurentina-Inviolatella Telecommunications Node	ICT	–
B: Fiumicino Airport	Transportation	Closure of check-in, ticketing, baggage services and transfers
C: ANSI Print Agency	ICT	Data transmission problem
D: Post Offices	Transportation	Delays and service perturbations
E: Banks	Finance	Delays and service perturbations
F: ACEA Power Distribution	Energy	Loss of SCADA monitoring and control
G: Fixed and Mobile Networks	ICT	Partial connectivity and loss of landline and mobile phone communications

network (node G) – landline communications as well as landline–mobile communications. In the example, the airport (CI_B) has a dependency risk $R_{A,B}$ from infrastructure CI_A . This risk value refers to the likelihood $L_{A,B}$ of the disruption of the telecommunications node to cascade to the airport as well as the societal impact $I_{A,B}$ on the airport when a failure has occurred at infrastructure CI_A , the source of the first-order dependency.

2.1.2. Risk of n th-order dependencies

Using the first-order dependencies as described in the example above, it is possible to assess the potential n th-order cascading risks using a recursive algorithm [20]. Let $\mathbb{C} = (CI_1, \dots, CI_m)$ be the set of infrastructures. Let $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow \dots \rightarrow CI_{Y_n}$ denote a chain of connected infrastructures of length n . Then, the recursive algorithm examines each critical infrastructure as the potential root of a cascading effect (denoted as CI_{Y_0}) and computes the dependency risk DR exhibited by CI_{Y_n} due to its n th-order dependence.

If $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow \dots \rightarrow CI_{Y_n}$ is a chain of dependencies, L_{Y_0, \dots, Y_n} is the likelihood of the n th-order cascading effect and I_{Y_{n-1}, Y_n} is the impact of the $CI_{Y_{n-1}} \rightarrow CI_{Y_n}$ dependency, then the cascading risk exhibited by CI_{Y_n} due to the n th-order dependency is computed as:

$$R_{Y_0, \dots, Y_n} = L_{Y_0, \dots, Y_n} \cdot I_{Y_{n-1}, Y_n} \equiv \prod_{i=0}^{n-1} L_{Y_i, Y_{i+1}} \cdot I_{Y_{n-1}, Y_n} \quad (1)$$

The cumulative dependency risk considers the overall risk exhibited by all the critical infrastructures in the sub-chains of the n th-order dependency. Let $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow \dots \rightarrow CI_{Y_n}$ be a chain of dependencies of length n . The cumulative dependency risk, denoted as $DR_{Y_0, Y_1, \dots, Y_n}$, is defined as the overall risk produced by an n th-order dependency:

$$DR_{Y_0, \dots, Y_n} = \sum_{i=1}^n R_{Y_0, \dots, Y_i} \equiv \sum_{i=1}^n \left(\prod_{j=1}^i L_{Y_{j-1}, Y_j} \right) \cdot I_{Y_{i-1}, Y_i} \quad (2)$$

Eq. (2) computes the overall dependency risk as the sum of the dependency risks of the affected nodes in the chain due to a failure realized in the source node of the dependency chain. The risk computation employs a risk matrix that combines the likelihood and incoming impact values of each vertex in the chain. Interested readers are referred to [19] for additional details about dependency risk estimation.

In many instances, the likelihood values are difficult to estimate or may not be available. This means that, while a dependency can be identified between two nodes, the probability of a failure to propagate between the two nodes is either unknown or certain (likelihood=1). In both cases, the following simplified version of Eq. (2), which follows the assumption that if a node fails, then the dependent nodes will also fail (likelihood=1), is used:

$$DR_{Y_0, \dots, Y_n} = \sum_{i=1}^n R_{Y_0, \dots, Y_i} \equiv \sum_{i=1}^n I_{Y_{i-1}, Y_i} \quad (3)$$

The n th-order dependency risk is then calculated as the cumulative impacts on the affected nodes in the dependency chain.

2.2. Fuzzy logic

The multi-risk methodology described above is static in time because Eqs. (1) through (3) are based on the maximum expected impact of each dependency. The values produced by these equations assume that (i) each dependency chain will always produce its worst case impact (and risk); and (ii) all the dependencies exhibit the same impact growth rate. However, in reality, neither all the critical infrastructure nodes in a chain escalate to their maximum consequences nor do they experience the same impact growth rate over time. For this reason, the multi-risk methodology is extended to incorporate a dynamic, time-based analysis and to assess partial failure scenarios. Fuzzy set theory is used to model this behavior.

Unlike classical set theory and classical logic, fuzzy set theory and fuzzy logic attempt to find approximations of vague groupings in order to project objective evaluations of values that are difficult to compute [29]. A fuzzy variable has a truth value in the range [0, 1] for a possible outcome. The goal is to approximate the time evolution of a cascading failure using fuzzy approximations of impact evolution for various growth models, similar to a real failure. For example, an incident might initially have a slow cascading effect on other dependent critical infrastructures and, as time goes by, a failure to restore operations might lead to catastrophic effects.

3. Time-based analysis of cascading and common-cause failures

First, the static dependency analysis methodology of [19–21] is extended to incorporate a dynamic time-based analysis model. Various cascading failure growth models are considered and fuzzy logic is applied to simulate realistic approximations of dynamic cascading failures. In addition, combined cascading and common-cause failures are considered when simulating the effects of dynamic, large-scale and major disasters.

3.1. Modeling time-based analysis of cascading failures

As mentioned above, I_{ij} and L_{ij} denote the impact (on a Likert scale) and the likelihood (as a percentage) of a failure experienced in dependency $CI_i \rightarrow CI_j$, respectively. These values are derived from assessments performed at the organization level by critical infrastructure owners/operators. As in most static risk assessment methodologies, the impact value I_{ij} refers to the maximum expected impact (in the worst-case scenario) regardless of the time taken for the maximum impact to be fully realized after a failure.

The following steps are used to perform dynamic time-based analysis:

1. *Model definition*: Define various failure growth rates.
2. *Setup*: Using the growth rates, pre-compute all the possible expected time-based impact values.
3. *Calculate fuzzy time-based impact values*: For a given dependency risk graph, use the pre-computed expected time-

based impact values as input to the fuzzy model in order to obtain a fuzzy approximation of the time-based impact for each dependency.

4. *Assess time-related dependency risks:* For each dependency chain, output the time-based cumulative dependency risk using the fuzzy time-based impact values.

These steps are described in the following subsections.

3.1.1. Model definition

Let T_{ij} denote the time period that the dependency $CI_i \rightarrow CI_j$ exhibits its maximum expected impact I_{ij} and let G_{ij} denote the expected growth of the failure (e.g., slow, linear or rapid) due to the dependency. The values of T_{ij} and G_{ij} , along with I_{ij} and L_{ij} , are obtained from critical infrastructure risk assessments. Finally, let t denote an examined time period after a failure.

In the remainder of this section, if there is no ambiguity, the dependency indices are omitted for simplicity. Thus, I , T and G are used instead of I_{ij} , T_{ij} and G_{ij} , respectively.

All the values are assigned from the following Likert scales:

- $I \in [1..9]$, where 1 is the lowest impact and 9 is the highest impact.
- $T, t \in [1..10]$, which is a granular time scale that uses the unavailability time periods: 1=15 min, 2=1 h, 3=3 h, 4=12 h, 5=24 h, 6=48 h, 7=1 w, 8=2 w, 9=4 w and 10= more than 4 w.
- $G \in [1..3]$, where 1 represents slow, 2 linear and 3 rapid growth (evolution) of a failure experienced due to the examined dependency.

Definition 1. Let $CI_i \rightarrow CI_j$ be an examined dependency with the maximum expected impact I experienced at time period T after a failure and let G be the growth evolution of the failure. The expected time-related impact of the dependency experienced at time t is computed as:

$$I(t) = \begin{cases} I^{(t/T)} & \text{if } G = 1 \text{ (slow evolution)} \\ I \cdot \left(\frac{t}{T}\right) & \text{if } G = 2 \text{ (linear evolution)} \\ I \cdot \log_T t & \text{if } G = 3 \text{ (rapid evolution)} \end{cases} \quad (4)$$

Obviously, $I(t) = I$ for $t \geq T$ if no mitigation controls are taken at the nodes.

Using Eq. (4), each dependency chain can be modeled with the most appropriate growth evolution and a different model can be used for each value of G . Rapid cascading effects are considered to escalate logarithmically (i.e., rapid short-term growth that stabilizes the maximum expected impact). Linear cascading effects are considered to escalate following a typical linear equation. Slow cascading effects follow an exponential growth rate that starts at low initial values and escalates towards the end of the time scale. Fig. 1 compares various growth rates for the same I and T values.

To clarify the main concepts, consider a dependency risk graph where the critical infrastructure dependencies

associated with a nuclear power facility are modeled as having very high impact ($I=9$) experienced during a relatively short time period ($T=3$ h) with rapid evolution ($G=3$). In the same graph, the edges starting from another typical energy provider are modeled with an impact value of 5 experienced 48 h after the failure and with a linear evolution ($G=2$). The model presented in this paper can project the evolution of the dependency risk paths for all the time periods for all the models.

Algorithm 1:.

```

procedure CALCULATEVALUES(T, I, G)
Inputs:
Time (min): T = {15, 60, 180, 720, 1,440, 2,880, 10,080, 20,160, 40,320, 60,480}
Impact: I = {1, 2, 3, 4, 5, 6, 7, 8, 9}
Growth: G = {slow, linear, rapid}
Result:
All I(t), ∀G ∈ G, I ∈ I and t, T ∈ T
for e0 in G do
Set e0 as G
for e1 in T do
Set e1 as T
for e2 in T do
Set e2 as t ▷ /* For t > T always output maximum impact */
if t > T then
I(t) = I
else if t ≤ T then
for I in I do
if G is rapid then
I(t) = I · log_T t
else if G is linear then
I(t) = I · (t/T)
else if G is slow then
I(t) = I^(t/T)
end if
end for
end if
end for
end for
end procedure

```

3.1.2. Setup

The growth rates computed using Eq. (4) are used to pre-compute all possible values of $I(t)$ for all possible combinations of I , T and G (see Algorithm 1). The output of Algorithm 1 is used as input to a fuzzy ranking system that provides realistic assessments of the evolution of potential failures.

The fuzzy logic classification system uses the following membership sets:

- *Impact set:* This set partitions the [1..9] impact scale as Very Low, Low, Medium, High and Very High: $\{VL = \{1\}, L = \{2, 3\}, M = \{4, 5\}, H = \{6, 7\}, VH = \{8, 9\}\}$.
- *Time set:* This set partitions the [1..10] time scale as Early, Medium, Late and Very Late: $\{E = \{1, 2, 3\}, M = \{4, 5, 6\}, La = \{7, 8\}, VLt = \{9, 10\}\}$.

In order to support the fuzzy mechanism (described below), the output of Algorithm 1 is stored in the form of pre-computed tables. The tables provide the expected time-related impact values for all possible values of G and T . All the tables are available at <http://github.com/geostergioip/CIDA/wiki>.

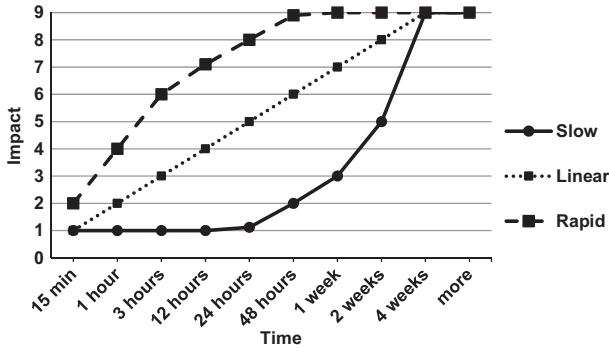


Fig. 1 – Expected impact evolution for various growth rates with $I=9$ and $T=4$ weeks.

Table 2 presents all possible time-based impact values $I_{ij}(t)$ for a rapidly-evolving failure ($G=3$) that experiences the worst-case impact at time $T=12$ h. The worst-case impact value of each column is assigned to cells in the row ($T=12$ h) and, obviously, to all rows below this row because the worst-case impact has already been realized. The time-based impact values for all the rows above the row ($T=15$ min up to $T=3$ h) are calculated by applying an inverse growth rate on the cell values in the $T=12$ h row using Eq. (4) for $G=3$.

Note that the impact and time values are grouped according to the fuzzy impact and time membership sets, respectively. For example, in Table 2, the fuzzy impact membership set Low contains impact values from 1 up to 3. All the values were obtained by applying the rapid (logarithmic) growth scale and time $T=12$ h as the expected time of occurrence of the worst-case impact value.

The output of the algorithm is computed during setup and stored in 30 tables. Specifically, for each of the three growth rates G , it is necessary to pre-compute and store one table for each of the ten possible T values.

3.1.3. Calculating fuzzy time-based impact values

The pre-computed tables containing all the expected time-based impact values $I(t)$ enable the fuzzy estimation of the time-related impact values for a dependency risk graph. For each dependency, the growth rate G and the expected time T of the worst-case impact value I are used to select the corresponding table from the database. Next, the fuzzy sets corresponding to the impact labels (Very Low, Low, Medium, High, Very High) are generated using the corresponding columns in the selected table. These fuzzy sets and linguistic IF-THEN rules are used to calculate the fuzzy value of the expected time-based impact value. The linguistic rules are expressed as: IF variable is property THEN action. All the IF-THEN rules are invoked using the constructed membership sets as linguistic variables to determine the fuzzy time-based impact value, which is given by:

$$\text{Fuzzy}(I, G, T, t) = I(t) \quad (5)$$

$I(t)$ is computed as follows. The initial processing stage invokes the appropriate IF-THEN rules and generates a result for each rule. The results are then combined to output a set of truth values. Each IF-THEN result is, essentially, a membership function and truth value that control the output set (i.e.,

the linguistic variables impact and time). The final step in obtaining a single quantitative value from a fuzzy set is known as “defuzzification.”

In the defuzzification process, all the IF-THEN output results are combined to yield a single fuzzy time-based impact value for each time point in the time scale. The rightmost membership defuzzification technique [50], which outputs the rightmost (i.e., highest) impact value, is employed. This is consistent with risk-based standards that tend to favor worst-case scenarios.

The output fuzzy time-base impact values $I(t)$ are considered to be more objective approximations of the expected impact at a given time because, instead of simply using the appropriate pre-computed expected time-based impact $I(t)$, the fuzzy values also consider their neighboring values. Thus the fuzzy values tend to better approximate real-world situations. In short, each dependency has its own expected growth, but it is also affected by the growth of its nearby dependencies.

As an example, suppose a dependency has input data $G=3$ and $T=12$ h. Thus, the fuzzy mechanism selects Table 2. The label Low of the fuzzy impact set (columns 3 and 4) contains only values 1, 2, 3. Using the aforementioned mechanism, the fuzzy membership set Low is defined as $\{(1, 0.05) (2, 0.55) (3, 0.4)\}$, where the second value of each tuple is the membership value of the corresponding impact value. The following subset of rules is used to calculate the Low output set of the value $I(t)$:

- Rule 17: IF Impact is Low AND Time is Early THEN Fuzzy_Impact is Very Low
- Rule 18: IF Impact is Low AND Time is Medium THEN Fuzzy_Impact is Medium
- Rule 19: IF Impact is Low AND Time is Late THEN Fuzzy_Impact is High
- Rule 20: IF Impact is Low AND Time is Very Late THEN Fuzzy_Impact is High

Next, the fuzzy impact for $I=2$ is computed. The fuzzy set that is most characterized by this value is Low. Assume that the worst-case scenario is at $T=12$ h. Based on Table 2, this time value belongs to the Medium time fuzzy set. Thus, using Rule 18, a Medium fuzzy time-based impact value is produced. Finally, the rightmost membership defuzzification technique is used on all the fuzzy sets to obtain the discrete time-based impact value.

3.1.4. Time-related multi-order dependency risk

The static model described in Section 2.1 can now be extended to provide multiple estimates of the evolution of the dependency risk. This is accomplished by replacing the static impact values I_{ij} with the dynamic fuzzy time-related impact values $I_{ij}(t)$ described above. These values are used to extend Eqs (2) and (3) and compute the n th-order dependency R_{Y_0, \dots, Y_n} and cumulative risk DR_{Y_0, \dots, Y_n} of a risk graph for each point on the time scale as follows:

$$DR_{Y_0, \dots, Y_n}(t) = \sum_{i=1}^n R_{Y_0, \dots, Y_i} \equiv \sum_{i=1}^n \left(\prod_{j=1}^i L_{Y_{j-1}, Y_j} \right) \cdot I_{Y_{i-1}, Y_i}(t) \quad (6)$$

or, if the likelihood assessments are omitted:

Table 2 – Pre-computed table with the expected time-related impact values for rapidly-evolving failures ($G=3$) with the worst impact at $T=12$ hours.

	Time Related Impact	Very Low (1)	Low (2)	Low(3)	Medium (4)	Medium (5)	High (6)	High (7)	Very High (8)	Very High (9)
Early	15 minutes	1	1	2	2	2	3	3	3	4
	1 hour	1	2	2	3	3	4	4	5	6
	3 hours	1	2	3	3	4	5	5	6	7
Medium	12 hours	1	2	3	4	5	6	7	8	9
	24 hours	1	2	3	4	5	6	7	8	9
	48 hours	1	2	3	4	5	6	7	8	9
Late	1 week	1	2	3	4	5	6	7	8	9
	2 weeks	1	2	3	4	5	6	7	8	9
Very Late	4 weeks	1	2	3	4	5	6	7	8	9
	> 4 weeks	1	2	3	4	5	6	7	8	9

$$DR_{Y_0, \dots, Y_n}(t) = \sum_{i=1}^n R_{Y_0, \dots, Y_i} \equiv \sum_{i=1}^n \mathcal{I}_{Y_{i-1}, Y_i}(t) \quad (7)$$

Eqs. (6) and (7) produce ten different values, one for each examined value of t .

3.2. Combining cascading and common-cause failure risks

The dependency risk values computed by Eqs. (6) or (7) assume a single initiating event (disruption) at a single critical infrastructure that results in cascading disruptions. It does not cover common-cause failures that simultaneously affect several, seemingly independent critical infrastructures. Such events can cause multiple cascading chains where the impact is introduced to multiple nodes in the graph simultaneously. Thus, the model must be extended to also capture failures that are simultaneously cascading and common-cause failures. A variety of incidents can serve as initiating events, including accidents, natural disasters and human-initiated attacks. For example, a common-cause initiating event may concurrently affect critical infrastructures (not identified as being directly dependent on each other) due to their physical proximity. Examples include a flood or national strike.

Let L_e be the likelihood of an event (threat) e . In the case of a natural disaster, the value of L_e can be assessed based on statistics of previous incidents, prognostics and the presence of vulnerabilities. However, the likelihood of an adversarial attack is more complex; in this case, the likelihood is affected by the motivation and skills of the adversary as well as his perceived impact of the attack. For this reason, expert opinions are commonly elicited and a worst-case approach is used to obtain the maximum valuation of risk.

Eq. (6) (or its simplified impact-only version, Eq. (7)) is used to evaluate all possible n th-order dependency risks. Let $\mathbb{C}\parallel$ be the set of all the examined critical infrastructures. The combined common-cause risk $CR(CI_{Y_0}, e)$ of all possible chains of cascading events $CI_{Y_0} \rightarrow CI_{Y_1} \rightarrow \dots \rightarrow CI_{Y_n}$ initiated by a common-cause failure event e for each possible source infrastructure $CI_{Y_0} \in \mathbb{C}\parallel$ can be computed as the sum of all possible risk chains $DR_{Y_0, \dots, Y_n} \forall Y_0 \in \mathbb{C}\parallel$ multiplied by the likelihood L_e of each examined event e :

$$CR(\mathbb{C}\parallel, e) = L_e \cdot \sum_{Y_0 \in \mathbb{C}\parallel} DR_{Y_0, \dots, Y_n}(t) \quad (8)$$

Every critical infrastructure (node) that is affected by a common-cause event e is examined as a possible root of a dependency chain (as CI_{Y_0}). For each CI_{Y_0} , the cumulative dependency risk DR is computed by applying Eq. (6) or its simplified version, Eq. (7).

4. CIDA tool

This section describes the design and implementation of the Critical Infrastructure Dependency Analysis (CIDA) tool [42], which implements the dependency analysis methodology described in this paper.

4.1. Neo4J graph database

A graph database model was selected as the main building block of CIDA. Graph databases are storage systems that provide index-free adjacency. Graph databases model data more effectively than relational databases, especially when the relationships between elements are the driving force for data model design [40,52]. In a graph database, every node only needs to know the nodes to which it is connected (i.e., its edges). This enables a graph database system to leverage graph theory to efficiently examine the connections and degrees of node connectivity. In addition, an edge utility enables a graph database to find results in associative data sets. Graph databases scale naturally to large data sets and to data sets with frequently-changing or on-the-fly schema [52].

The Neo4J [28] framework was selected to implement CIDA due to its adaptability, scalability and efficiency. According to recent empirical studies [2,16,40], Neo4J outperforms other systems in load time for thousands of elements as well as in the time required to compute the total paths and detect the shortest path. On the other hand, Neo4J has inferior performance for highly volatile network topologies (e.g., graphs with frequent changes in nodes and edges) compared with other graph model approaches (e.g., DEX and Titan-Cassandra). This deficiency is not relevant to this work because critical infrastructure dependencies do not change frequently.

Neo4J builds on the property graph model: nodes may have labels and each label can serve as an informational entity. The nodes are connected via directed, typed relationships. Both

nodes and relationships can hold arbitrary properties (key-value pairs). Although there is no rigid schema, the node labels and relationship types can provide to the nodes as much meta-information as necessary for the node attributes required by a specific schema. When importing data into a graph database, the relationships are treated with equal importance as the database records themselves [52].

Neo4j deploys a single server instance that can handle a graph of billions of nodes and relationships. If the data throughput exceeds a limit imposed by the computing resources, the graph database can be distributed among multiple servers to provide a scalable configuration with high availability. In addition, Neo4j listeners can capture useful signals (notices) that objects broadcast in response to certain events (e.g., changes to property values or user interactions). As explained later, these two graph functions are the most important factors that affect the computational time for analyzing dependent and interconnected critical infrastructures. During development, the Blueprints technology was also used; this provides a property graph model interface that supports a Neo4j graph database.

4.2. Dependency graph analysis

The CIDA graph-based critical infrastructure dependency analysis tool implements the methodology presented in this paper in order to dynamically analyze the risk of critical infrastructure dependency chains under cascading and/or common-cause failures. CIDA computes the security risk and/or impact evolution of dependencies over time. CIDA represents complex graphs of thousands of dependent critical infrastructures as a weighted, directed graph. The weight of each connection (edge) between two critical infrastructures is the (maximum) estimated dependency risk value derived from the dependency between the two infrastructures. To render the computation of the risk dependency paths more efficient, a maximum depth limit on the examined dependencies is set. Empirical research [49] has revealed that cascading effects rarely affect critical infrastructures beyond fifth-order dependencies. Thus, the fifth-order is used as the upper limit on the critical infrastructure dependencies that are evaluated.

CIDA takes as input the nodes and edges of a graph (either from a spreadsheet or via the graphical interface). For each edge $CI_i \rightarrow CI_j$, the estimated likelihood L_{ij} and the (maximum) expected impact I_{ij} are used to conduct a static analysis. For a dynamic time-based analysis, the expected time T_{ij} of the maximum impact and the expected growth rate G_{ij} for the dependency are also required.

Given an input dependency graph, CIDA yields the following outputs:

- A table of all existing dependency paths up to a maximum dependency order (fifth-order by default).
- In a static analysis (based only on L_{ij} and I_{ij}), for each dependency path, the cumulative dependency risk is computed using Eq. (2).
- In a dynamic analysis (additionally, using the time-related inputs T_{ij} and G_{ij}), the expected cumulative dependency risk is computed for various time frames using Eq. (6).

- The dependency paths may be sorted based on their cumulative dependency risk values and can also be presented in a graphical form. If the user has set a maximum risk threshold, CIDA also highlights all the paths that have values greater than the risk threshold. The paths can be exported to a spreadsheet or an XML file for further analysis.
- CIDA graphically highlights the path that exhibits the maximum cumulative dependency risk using a modified version of Dijkstra's algorithm, which uses negative weights (risk values) to compute the maximum weighted path.

Based on the user's preferences, CIDA can compute cyclic paths (if feedback effects are considered) as well as acyclic paths (if feedback effects are excluded). In cases where the likelihood values are not available, CIDA can proceed using only the impact values and Eqs. (3) and (7) can be used for static and dynamic analyses of dependency impact paths, respectively. Obviously, impact estimates will be higher than risk estimates; this is because they represent worst-case scenarios that assume if a node fails, then all the following nodes will experience total failure.

4.3. Modeling infrastructures and dependencies

In CIDA, each node may represent a critical infrastructure or an "autonomous" sub-component of a critical infrastructure (e.g., a power generation substation) depending on the desired level of analysis. Each node in a graph supports the following attributes:

- *Name*: A unique name for the critical infrastructure node.
- *Critical infrastructure operator*: The critical infrastructure operator responsible for the node. If the analysis is performed at the unit level, then one operator may be responsible for several nodes.
- *Critical infrastructure sector*: The specific critical infrastructure sector to which the node belongs (e.g., information and communications technology).
- *Critical infrastructure sub-sector*: The specific sub-sector to which the node belongs (e.g., telecommunications).
- *Node location*: The latitude and longitude of the location. This captures geographical dependencies between critical infrastructures and helps evaluate potential threats that concurrently affect multiple nodes.
- *MaxPath*: A Boolean value that indicates if the node belongs to the maximum risk path.

CIDA supports seventeen critical infrastructure sectors, including information and communications, energy, transportation, water systems and the critical infrastructure sectors identified in [6,46]. CIDA also supports the modeling of all types of dependencies. While logical, informational and physical dependencies may be defined in service level agreements and are, thus, easier to identify, geographical dependencies may be missed. CIDA can automatically identify geographical dependencies based on the locations provided for the critical infrastructure nodes. This enables CIDA to study threat scenarios within a specific geographical region and, thus, to assess the effects of geographical dependencies.

CIDA computes the risk for each individual dependency path using the appropriate equations described above. The risk values are then associated with the weighted, directed graph stored in the Neo4J database. CIDA provides a visualization interface using the JUNG2 graph visualization library that is supported by Blueprints and Neo4J.

Fig. 2 shows an example dependency risk graph output. Each node represents a critical infrastructure and the weight of each edge is the estimated dependency risk value that derives from the dependency between two critical infrastructures. The type of the dependency is also depicted. Dark grey edges and nodes indicate the maximum cascading risk path. The complete set of paths and the relative risk values may be exported to a spreadsheet for further analysis.

5. Efficiency analysis

This section empirically analyzes the efficiency of the CIDA dependency analysis tool using granular random scenarios containing from 10 to 1000 critical infrastructure nodes as test cases. All the tests were performed using a workstation with an Intel Core i-7 2.7 GHz processor with four cores and 16 GB RAM. For each scenario, two cases with different degrees of connectivity were examined. In the low connectivity case, each node was randomly connected to at least one and at most three other nodes. In the high connectivity case, each node was randomly connected to at least four nodes and at most five other nodes. For each case, the execution time was multiplied by a factor of ten to estimate the execution time for dynamic analysis. This is because, in the case of time-based analysis, all the computations are repeated for each examined time frame.

In real-world scenarios the majority of nodes would have connectivity degrees no greater than three and only a small fraction of nodes would have high degrees of connectivity of four or five (e.g., major electrical substations and key telecommunications nodes) [49]. Thus, it is reasonable to expect that, in a real-world scenario, the average connectivity degree of nodes is between these values and the expected execution time is between the two execution times (for the same number of nodes).

Each test was repeated ten times and the mean execution time was computed. The execution time included the time for computing all dependency risk paths up to fifth-order dependencies and the time required to sort the paths based on their execution times and compute the maximum dependency risk path. Again, since cascading effects rarely affect nodes more than four hops away from the source, the computation of up to fifth-order dependencies was adequate to cover the majority of the cases [49]. As expected, the computational time of the complete set of dependency risk paths increases exponentially with the number of nodes (Fig. 3). Also, the connectivity degree appears to affect the execution time significantly. For example, the 1000 node scenario required about 20 min for nodes with low connectivity (one to three edges per node) and more than 332 min for high connectivity nodes (four to five edges per node).

Since the maximum path problem is NP-complete, it is obvious that CIDA cannot provide a complete theoretical

solution. However, with reasonable limitations in system parameters such as the maximum number of nodes, degree of node connectivity and degree of dependency order, CIDA can efficiently provide useful results for large-scale and cross-sectoral real-world scenarios.

6. Analyzing real-world scenarios with cascading effects

This section evaluates scenarios based on a real-world case involving cascading effects to demonstrate the applicability of CIDA. Although the tests are based on a real case, the impact, likelihood and time-related inputs assigned to each dependency are not based on real risk assessment results. Such assessment results are not publicly available, so the tests rely on subjective assumptions for demonstration purposes.

6.1. Real-world cascading blackout

This scenario, which comprises nine nodes, is based on the well-known electricity blackout in California [33]. The scenario was selected because it is well-documented and exhibits several complex cross-sectoral and multi-order cascading dependencies. The initiating event in the scenario was the failure of an electric power substation (node A in Fig. 4). The event triggered several first-order dependencies: disruption of a natural gas production infrastructure (node B), disruption of petroleum pipelines (node E) that transported jet fuel to neighboring states and disruptions of massive water pump units (node H).

As shown in Fig. 4, the disruption of gas production (node B) directly impacted gas supplies for steam injection units (node C), a second-order dependency. The steam injection units affected the operation of heavy oil recovery units (node D), a third-order dependency, further exacerbating power problems at node A (feedback loop). Similarly, the disruption of petroleum pipelines (node E) caused inventories to build up at refineries and draw down at product terminals, including at several major California airports (node F), a second-order dependency. The reduction of jet fuel stocks at the airports caused several major airline operators (node G) to consider contingency plans, a third-order dependency. Finally, the disruption of water pump units (node H) affected crop irrigation at several fields (node I), a third-order dependency.

Table 3 presents the input values used in the scenario. For each dependency, a likelihood and maximum expected impact estimate are provided. Also, the expected time taken for the maximum impact to be manifested is provided, along with an estimate of the growth rate of the failure.

Note that the input data for first-order dependencies can be fed to CIDA via a spreadsheet (Table 3) or via a graphical interface (Fig. 5). Given the input data, CIDA computes the complete set of dependency risk paths in a time frame for each dependency chain of order no greater than five using Eq. (6).

CIDA outputs a graphical representation of the examined dependency risk graph (Fig. 6). In this case, the graph of the nine examined nodes produces 38 chains with orders ranging

from two to five and with potential risk values between 1.5 and 13.17. The nodes and edges marked with darker colors are associated with the maximum cumulative dependency risk path.

6.2. Cascading-only dependency failure scenario

After CIDA has evaluated all the dependency risk paths, it is possible for a user to examine several scenarios in an efficient manner. One scenario is to analyze and compare all possible cascading effects. CIDA produces a list of all the dependency paths sorted according to their cumulative dependency risk values. This helps the user to identify all the potential dependency risks that are above a specified threshold value. For example, Fig. 7 shows the subset of dependency risk paths that exhibit cumulative risk values above a threshold of five, regardless of the time taken to reach the threshold. It also shows the risk values of the maximum risk paths for all the examined time periods as well as the maximum risk levels for the remaining paths at the time of occurrence. Note that the user may project different paths using different thresholds.

The threshold parameter assists users in determining the most effective risk mitigation strategies. In Fig. 7, it is easy to see that the four dependency paths with the highest risk values surpass the threshold within one hour after the initial failure and they all start at node A. Thus, a cost-effective strategy would start by applying mitigation controls at node A with a rapid response time; this would decrease the overall dependency risk substantially. If mitigation controls at node A are too expensive, an alternative strategy is to reduce the likelihood of cascades to the most important second-order dependencies of node A (nodes B and E).

A second result of the analysis is that, although path A–B–C–D exhibits the highest risk for almost all the examined time frames, the graph reveals that path A–E–F–G is the most critical

path about 12 h after a cascading failure. This is because, although dependencies A–B and A–E both have rapid growth, the second dependency is expected to have the fastest convergence to its maximum impact ($T_{AE} = 12$ h). Recall that this methodology supports different T_{ij} and G_{ij} values for each dependency.

A third result can be obtained by comparing the evolution of sub-paths that exhibit high risk. For example, although the path A–B–C–D is the highest risk path, its sub-path A–B–C already exhibits an impact greater than the threshold within 1 h. Thus, it is necessary to implement mitigation controls at the first- or second-order dependency.

Finally, for the remaining dependency paths A–H–I, B–C–D–A–E–F and C–D–A–E–F–G, it is safe to consider mitigation controls with slower response times because they have relatively low risk values even for long time frames. Note that the last two paths include the chain A–E–F as a sub-path, so mitigation controls implemented at A–E–F concurrently reduce the risk of four of the seven most critical paths.

6.3. Combined common-cause and cascading scenarios

Another scenario is to analyze all the cascading effects that may potentially be triggered by a common-cause failure. In this case, for each examined node, CIDA computes the sum of the dependency risks of all the existing distinct paths originating from the node (to avoid repetitions of the same risk chain, only distinct paths are considered). In the examined case, node A (as expected) is by far the most critical node for common-cause failures – the sum of distinct risk paths is 33.22 (paths A–B–C–D, A–E–F–G and A–H–I). In a common-cause scenario, at least two nodes are directly affected by the initiating event, which serves as the common cause of the failures. Therefore, for each affected node, it is necessary to calculate the sum of distinct risk paths and weight these values with the likelihood value L_e of the initiating event e (failure or attack) that may be realized at the source nodes. The combined risk for every possible initiating event e at each directly affected node is computed using Eq. (8).

Note that the complete set of dependency chain risks is already an output of the CIDA tool. Thus, the evaluation of the possible common-cause failures is based on “ready-to-use” risk chains. The user may add initiating events and likelihood values to every node according to the probability that the examined event causes a failure to the node. The initiating events e and likelihood values L_e are selected by the user based on expert opinion and statistical data. Obviously, it is reasonable to first examine nodes that exhibit higher sums of risk values before they are weighted with L_e .

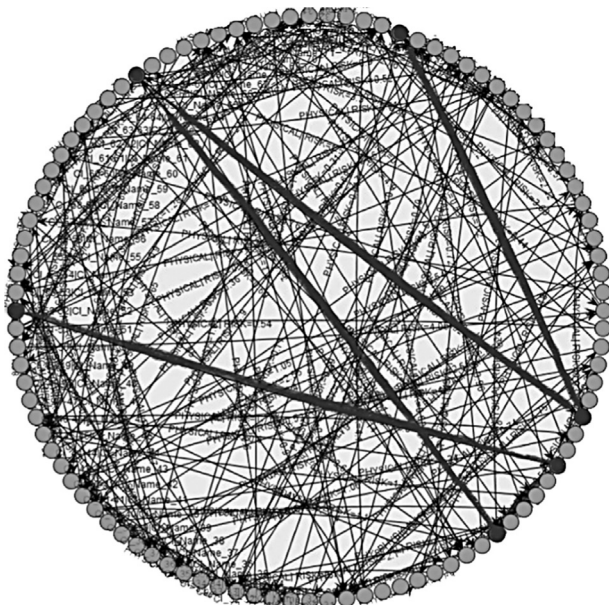


Fig. 2 – Dependency risk graph for a test scenario with 100 nodes.

7. Comparison with other approaches

Ouyang [30] distinguishes modeling and simulation approaches as: (i) empirical; (ii) agent based; (iii) system dynamics based; (iv) economic theory based; (v) network (topology or flow) based; and (vi) others (hierarchical holographic modeling based, high level architecture based, Petri net based, dynamic control system theory based, Bayesian network based, etc.). Another taxonomy of dependency

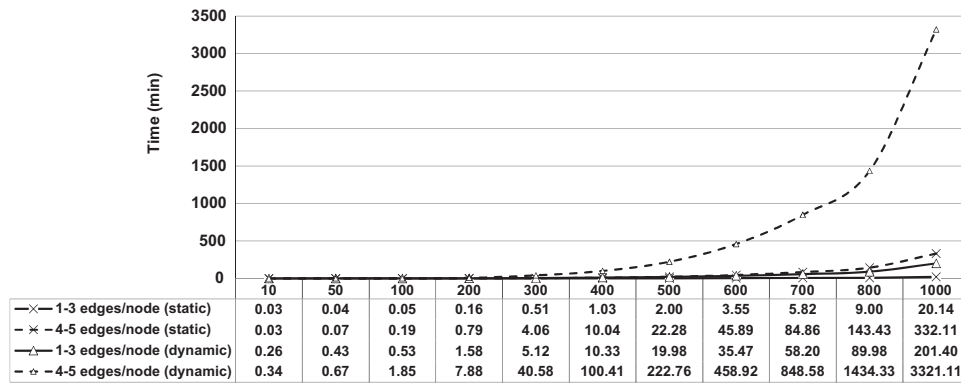


Fig. 3 – Execution time versus number of nodes.

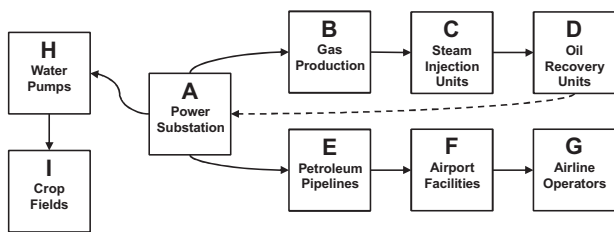


Fig. 4 – Dependencies in the California blackout scenario.

Table 3 – Input values for the blackout scenario.

CI_i	CI_j	L_{ij}	I_{ij}	T_{ij}	G_{ij}
A	B	0.90	8	24 h	Rapid
B	C	0.65	6	48 h	Linear
C	D	0.70	6	48 h	Slow
D	A	0.15	3	2 w	Slow
A	H	0.80	6	12 h	Rapid
H	I	0.65	7	48 h	Linear
A	E	0.90	8	12 h	Rapid
E	F	0.70	5	24 h	Slow
F	G	0.25	8	1 w	Slow

models [24,55] has six broad categories: (i) aggregate supply and demand tools; (ii) dynamic simulations; (iii) agent based models; (iv) physics based models; (v) population mobility models; and (vi) Leontief input-output models. The methodology presented in this paper can be categorized as hybrid because it has characteristics of empirical methods (as a risk based approach) and network based methods (as a graph modeling tool).

Empirical methods have been criticized by researchers due to the lack of statistical data required to assess the likelihood of potential events. While probability data may be difficult to collect for many critical infrastructures, efforts have already been made to do so in specific critical infrastructure sectors. For example, Carreras et al. [5] have conducted statistical studies of blackouts that enable the identification of critical power lines or groups of power lines for a given network model. This helps identify critical clusters of lines that are likely to trigger or propagate cascading effects due to power line vulnerabilities.

The approach presented in this paper also draws from network based methods in that it combines a method for discovering dependency risk paths with an automated modeling and analysis tool. It enables the dependencies of the connected infrastructures to be depicted as a graph and critical paths to be identified. Such flow based network approaches are described in the literature. They either model the flow of products or services between critical infrastructures in a uniform model [25,43,44] or they combine various sector-based flow models [31,36].

Most modeling, simulation and analysis tools in the literature are sector-specific. For example, in the water sector, OpenMI [45] supports federated modeling and simulation for a wide range of technical, organizational and economic aspects related to water systems (e.g., sea, dikes,

groundwater, water management and more). Other approaches allow for integrated or federated simulations that combine models from multiple sectors; examples include DIESIS [35,47], EPIC [41] and I2Sim [27].

With regard to the level of analysis, CIDA does not model infrastructures at the component level. For this reason, CIDA has similarities with the empirical Leontief input-output models used for high-level multi-sectoral risk assessments [15,37-39]. These approaches measure the dependencies existing between critical infrastructure sectors in terms of economic relationships. The approach of Setola et al. [39] considers domain expert opinion and uses fuzzy logic to assess the impact induced by direct and higher-order dependencies between critical infrastructures. The approach assumes that critical infrastructure operators (or the experts who conduct assessments) provide input data about the impact values for resource outages of various durations in each critical infrastructure. Like CIDA, the approach of Setola et al. [39] uses fuzzy logic. The approach of Setola et al. [39] uses it to minimize the uncertainty and ambiguity associated with subjective information received from domain experts. On the other hand, CIDA combines fuzzy logic with various time growth models. Each dependency may follow a different growth rate and fuzzy logic is used to objectify the evolution of each dependency, taking into consideration the states of other nearby dependencies. This enables CIDA to output results for various time frames (also available in [39]) as well as for alternative failure growth rates. Moreover, CIDA uses a dependency risk graph to model a variety of dependencies, not just economic dependencies.

Other approaches [15,37,38] use the input–output inoperability model to assess the dependencies between various sectors of an economy and to forecast the effects of a disruption in one sector on another sector. However, the approach presented in this paper is not a purely economical one.

Another important difference is that CIDA allows alterative graphs to be created to analyze dependencies that occur in abnormal operating conditions; in contrast, the inputs to the approaches described in [15,37,38] only incorporate dependencies in normal economic operations. Additionally, CIDA can perform a time-based analysis, which offers different risk results according to the time frame studied and the rate at which the impact evolves in each critical

infrastructure. With regard to the input data required for analysis, the approaches use aggregated economic data on a sector basis, while CIDA uses risk assessment data provided by owners and operators for each infrastructure.

Another economic-based approach is implemented in N-ABLE, a NISAC tool [11]. N-ABLE is a large-scale microeconomic simulation tool that models complex supply chains, spatial market dynamics and critical infrastructure interdependencies between U.S. businesses. N-ABLE is to model how U.S. businesses adapt to and recover from disruptive events. CIDA, on the other hand, is not specifically engineered to model the economic impact at the microeconomic level.

The Critical Infrastructure Protection/Decision Support System (CIP/DSS) [4,7] enables decision makers to determine the consequences of infrastructure disruptions. CIP/DSS assesses uncertainties with regard to threats, vulnerabilities and consequences of terrorist acts and natural disasters at the metropolitan and national scales. It models interdependencies for all U.S. critical infrastructures and key resources and calculates the impacts that cascade into these interdependent infrastructures and into the national economy. The CIDA tool can benefit from the CIP/DSS representation of mitigation alternatives in order to enhance the selection of mitigation strategies.

Experiments reveal that CIDA can efficiently compute the risks of all the dependency risk paths when reasonable limits are placed on the order of dependencies. However, the execution times for large-scale scenarios comprising hundreds of nodes may not be feasible for real-time analysis and response. CIPR/Sim [53] is an effective real-time analysis tool that imports real-time data from numerous analysis modules, including a real-time digital simulator for electric grid analysis, QualNet for telecommunications analysis and PC Tide for wind speed and flood surge analysis.

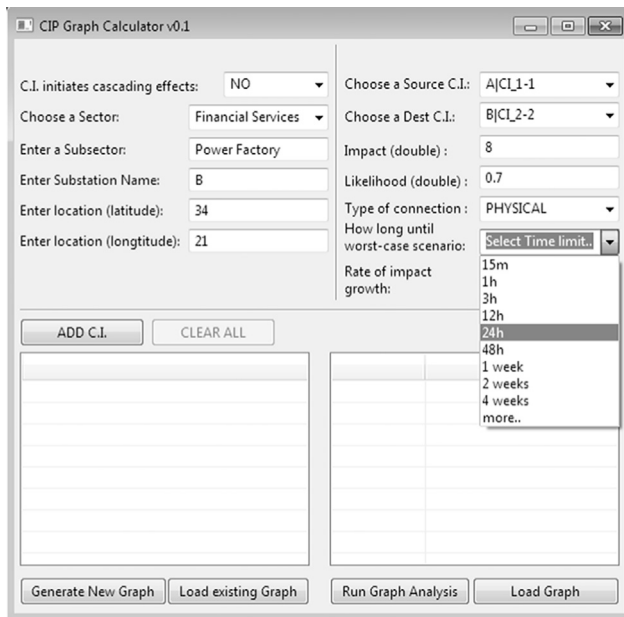


Fig. 5 – Graphical interface for inputting node and edge data.

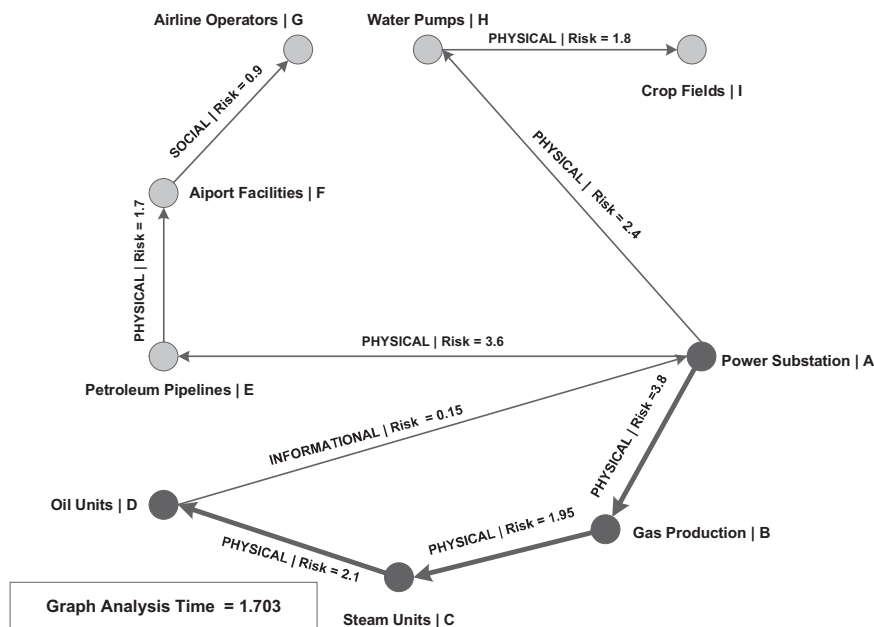


Fig. 6 – Dependency risk graph of the blackout scenario.

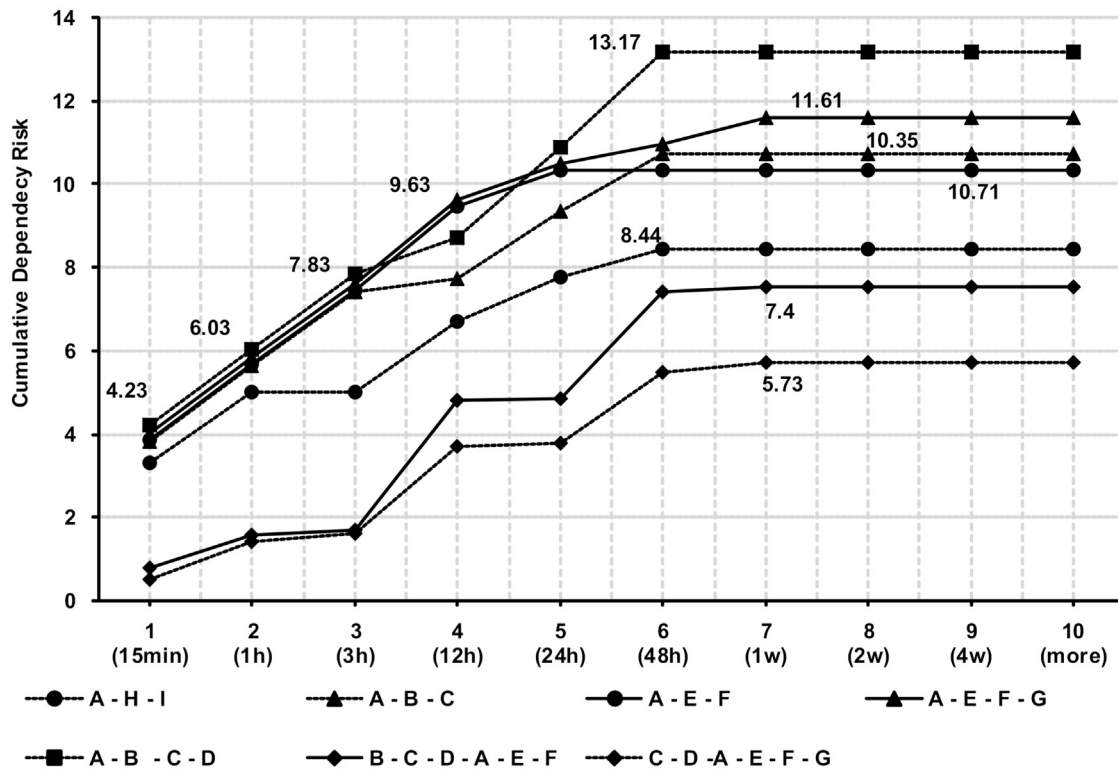


Fig. 7 – Dependency risk paths with cumulative dependency risk greater than a threshold of five.

8. Conclusions

The CIDA modeling and analysis tool supports the proactive study of large-scale dependency scenarios. In particular, the tool helps risk assessors and critical infrastructure protection decision makers to assess dependency risks before an initiating threat is realized. By analyzing the complete set of potential dependency paths, a user can project all the cascading effects that may be realized and flag dependency risks that are above a threshold for further attention.

CIDA can also be used to run specific scenarios of interest to risk assessors. While the computation of the complete set of dependency risk paths may provide useful information and reveal “hidden” dependency risks, assessors may also use CIDA to examine specific realistic scenarios. These include “what-if” scenarios that only consider initiating security events that affect one (or some) critical infrastructure nodes. Such a scenario may involve a major physical disaster that initially affects all the nodes in a geographical area. This is easily implemented because the attributes of each node in CIDA can incorporate geographical coordinates. Thus, it is possible to assess scenarios involving common-cause failures to targeted nodes and examine the cascading effects based on geographical dependencies.

The proactive assessment of risk mitigation controls helps increase resilience. Based on real input data, it is feasible to examine hundreds of scenarios, including previous incidents. Risk assessors may efficiently employ slight variations of dependency graphs with different weights and even different dependencies to simulate the implementation of alternative

risk mitigation controls. For example, if a particular path has been identified by exhaustive computations as a dependency risk path above the maximum risk threshold, CIDA can be used to project the effect of implementing redundant security controls to decrease the impact of a dependency. Alternatively, it can be used to reduce or eliminate dependencies using security controls in order to optimize the topology of interdependent critical infrastructures. Both examples increase the absorbing capacity of individual critical infrastructure nodes or a network of critical infrastructures to enhance the overall resilience.

The CIDA tool can also be used to identify and target key nodes in order to make them more resistant to failures or to improve their restorative capabilities. In this way, it is possible to evaluate the benefits of various alternative and/or complementary mitigation controls, and provide convincing arguments about the expected benefits of mitigation strategies. Note that studies of such scenarios can be parallelized using additional computing resources.

A limitation of the methodology presented in this paper is its reliance on prior risk assessments of critical infrastructures. This is inherent to all the empirical risk approaches – empirical risk-based approaches analyze dependencies based on previous incidents (historical incident or disaster data) coupled with expert opinion to identify alternative measures that minimize the dependency risk (see, e.g., [13,18,48]). It is highly unlikely for a single critical infrastructure owner or operator to have access to real data about other critical infrastructures. Thus, the methodology can only be applied at a higher layer. For example, sector coordinators or regulators may collect data about a specific sector such as energy or

information and communications technology. National critical infrastructure protection authorities may also be able to collect such information. This data can be gradually incorporated in the CIDA database to support the analysis of large-scale scenarios. Moreover, the CIDA tool can be used with the expected impacts of dependent critical infrastructures while ignoring likelihood parameters that are generally more difficult to collect [13]. Details of the entire CIDA Project, including the stress test implementation and the tool itself, are available at <http://github.com/geostergiop/CIDA>.

Future work will focus on enhancing CIDA with statistical data pertaining to potential initiating events and their likelihoods of occurrence for key sectors such as energy and information and communications technology. This will further assist risk assessors in evaluating combined common-case and cascading failures.

Acknowledgements

The research of Marianthi Theocharidou was performed under the CIPRNet Project funded by the European Union's Seventh Framework Programme for Research, Technological Development and Demonstration under Grant no. 312450. Some of the preliminary work was supported by the Excellence and Extroversion Programme (Action 2) of the Athens University of Economics and Business. The authors also wish to thank our colleague, Dr. David Ward, for his suggestions that have greatly improved the quality of this paper.

REFERENCES

- [1] G. Andersson, P. Donalek, R. Farmer, N. Hatziaargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, R. Schulz, A. Stankovic, C. Taylor and V. Vittal, Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance, *IEEE Transactions on Power Systems*, vol. 20(4), pp. 1922–1928, 2005.
- [2] S. Batra and C. Tyagi, Comparative analysis of relational and graph databases, *International Journal of Soft Computing and Engineering*, vol. 2(2), pp. 509–512, 2012.
- [3] S. Buldyrev, R. Parshani, G. Paul, H. Stanley and S. Havlin, Catastrophic cascade of failures in interdependent networks, *Nature*, vol. 464, pp. 1025–1028, April 15, 2010.
- [4] B. Bush, L. Dauelsberg, R. LeClaire, D. Powell, S. Deland and M. Samsa, Critical Infrastructure Protection Decision Support System (CIPS/DSS) Project overview, *Proceedings of the Twenty-Third International Conference of the System Dynamics Society*, pp. 17–21, 2005.
- [5] B. Carreras, D. Newman and I. Dobson, Determining the vulnerabilities of the power transmission system, *Proceedings of the Forty-Fifth Hawaii International Conference on System Sciences*, pp. 2044–2053, 2012.
- [6] Commission of the European Communities, Green Paper on a European Programme for Critical Infrastructure Protection, COM(2005)576 Final, Brussels, Belgium, 2005.
- [7] S. Conrad, R. LeClaire, G. O'Reilly and H. Uzunaloglu, Critical national infrastructure reliability modeling and analysis, *Bell Labs Technical Journal*, vol. 11(3), pp. 57–71, 2006.
- [8] Critical Infrastructure Preparedness and Resilience Research Network, CIPRNet, Fraunhofer IAIS, Sankt Augustin, Germany (<http://www.ciprnet.eu>), 2014.
- [9] Department of Homeland Security, NIPP 2013: Partnering for Critical Infrastructure Security and Resilience, Washington, DC, 2013.
- [10] L. Duenas-Ororio and S. Vemuru, Cascading failures in complex infrastructure systems, *Structural Safety*, vol. 31(2), pp. 157–167, 2009.
- [11] M. Ehlen and A. Scholand, Modeling interdependencies between power and economic sectors using the N-ABLE agent-based model, *Proceedings of the IEEE Power Engineering Society General Meeting*, pp. 2842–2846, 2005.
- [12] European Commission, The Post 2015 Hyogo Framework for Action: Managing Risks to Achieve Resilience, COM(2014)216 Final, Brussels, Belgium, 2014.
- [13] L. Franchina, M. Carbonelli, L. Gratta, M. Crisci and D. Perucchini, An impact-based approach for the analysis of cascading effects in critical infrastructures, *International Journal of Critical Infrastructures*, vol. 7(1), pp. 73–90, 2011.
- [14] R. Francis and B. Bekera, A metric and frameworks for resilience analysis of engineered and infrastructure systems, *Reliability Engineering and System Safety*, vol. 121, pp. 90–103, 2014.
- [15] Y. Haimes and P. Jiang, Leontief-based model of risk in complex interconnected infrastructures, *Journal of Infrastructure Systems*, vol. 7(1), pp. 1–12, 2001.
- [16] S. Jouili and V. Vansteenberghe, An empirical comparison of graph databases, *Proceedings of the International Conference on Social Computing*, pp. 708–715, 2013.
- [17] D. Judi, A. Kalyanapu, S. Burian, B. Daniel and T. McPherson, Wide-area flood inundation and infrastructure risk assessment simulation framework, *Proceedings of the Second IASTED International Conference on Water Resources Management*, 2007.
- [18] G. Kjolle, I. Utne and O. Gjerde, Risk analysis of critical infrastructures emphasizing electricity supply and interdependencies, *Reliability Engineering and System Safety*, vol. 105, pp. 80–89, 2012.
- [19] P. Kotzanikolaou, M. Theoharidou and D. Gritzalis, Assessing n-order dependencies between critical infrastructures, *International Journal of Critical Infrastructures*, vol. 9(1/2), pp. 93–110, 2013.
- [20] P. Kotzanikolaou, M. Theoharidou and D. Gritzalis, Cascading effects of common-cause failures in critical infrastructures, in *Critical Infrastructure Protection VII*, J. Butts and S. Sheno (Eds.), Springer, Heidelberg, Germany, pp. 171–182, 2013.
- [21] P. Kotzanikolaou, M. Theoharidou and D. Gritzalis, Interdependencies between critical infrastructures: Analyzing the risk of cascading effects, in *Critical Information Infrastructure Security*, S. Bologna, B. Hammerli, D. Gritzalis and S. Wolthusen (Eds.), Springer-Verlag, Berlin, Heidelberg, Germany, pp. 104–115, 2013.
- [22] R. Kozik and M. Choras, Current cyber security threats and challenges in critical infrastructure protection, *Proceedings of the Second International Conference on Informatics and Applications*, pp. 93–97, 2013.
- [23] W. Kroger, Emerging Risks Related to Large-Scale Engineered Systems, Technical Report, International Risk Governance Council, Lausanne, Switzerland, 2010.
- [24] W. Kroger and E. Zio, *Vulnerable Systems*, Springer-Verlag, London, United Kingdom, 2011.
- [25] E. Lee, J. Mitchell and W. Wallace, Restoration of services in interdependent infrastructure systems: A network flows approach, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37(6), pp. 1303–1317, 2007.
- [26] E. Luijff and D. Stolk, An international tabletop exercise on critical infrastructure protection: The lessons identified, *International Journal of Critical Infrastructures*, vol. 6(3), pp. 293–303, 2010.
- [27] J. Marti, J. Hollman, C. Ventura and J. Jatskevich, Dynamic recovery of critical infrastructures: Real-time temporal

- coordination, *International Journal of Critical Infrastructures*, vol. 4(1/2), pp. 17–31, 2008.
- [28] Neo Technology, Neo4j Graph Database, San Mateo, California (<http://www.neo4j.org>), 2014.
- [29] V. Novak, I. Perfilieva and J. Mockor, *Mathematical Principles of Fuzzy Logic*, Springer, New York, 1999.
- [30] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, *Reliability Engineering and System Safety*, vol. 121, pp. 43–60, 2014.
- [31] M. Ouyang and L. Duenas-Osorio, An approach to design interface topologies across interdependent urban infrastructure systems, *Reliability Engineering and System Safety*, vol. 96(11), pp. 1462–1473, 2011.
- [32] S. Panzieri and R. Setola, Failure propagation in critical interdependent infrastructures, *International Journal of Modeling, Identification and Control*, vol. 3(1), pp. 69–78, 2008.
- [33] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding and analyzing critical infrastructure interdependencies, *IEEE Control Systems*, vol. 21(6), pp. 11–25, 2001.
- [34] B. Robert, A method for the study of cascading effects within lifeline networks, *International Journal of Critical Infrastructures*, vol. 1(1), pp. 86–99, 2004.
- [35] E. Rome, S. Bologna, E. Gelenbe, E. Luijff and V. Masucci, DIESIS: An interoperable European federated simulation network for critical infrastructures, *Proceedings of the SISO European Simulation Interoperability Workshop*, pp. 139–146, 2009.
- [36] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. De Porcellinis and R. Setola, Modeling interdependent infrastructures using interacting dynamical models, *International Journal of Critical Infrastructures*, vol. 4(1/2), pp. 63–79, 2008.
- [37] J. Santos, Inoperability input-output modeling of disruptions to interdependent economic systems, *Systems Engineering*, vol. 9(1), pp. 20–34, 2006.
- [38] J. Santos and Y. Haimes, Modeling the demand reduction input-output (I–O) inoperability due to terrorism of interconnected infrastructures, *Risk Analysis*, vol. 24(6), pp. 1437–1451, 2004.
- [39] R. Setola, S. De Porcellinis and M. Sforma, Critical infrastructure dependency assessment using the input-output inoperability model, *International Journal of Critical Infrastructure Protection*, vol. 2(4), pp. 170–178, 2009.
- [40] B. Shao, H. Wang and Y. Xiao, Managing and mining large graphs: Systems and implementations, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 589–592, 2012.
- [41] C. Siaterlis, B. Genge and M. Hohenadel, EPIC: A testbed for scientifically rigorous cyber-physical security experimentation, *IEEE Transactions on Emerging Topics in Computing*, vol. 1(2), pp. 319–330, 2013.
- [42] Y. Stergiopoulos, P. Kotzanikolaou, M. Theocharidou and D. Gritzalis, CIDA: Critical Infrastructure Dependency Analysis Tool, Information Security and Critical Infrastructure Protection Laboratory, Department of Informatics, Athens University of Economics and Business, Athens, Greece (<http://github.com/geostergiop/CIDA>), 2014.
- [43] N. Svendsen and S. Wolthusen, Analysis and statistical properties of critical infrastructure interdependency multi-flow models, *Proceedings of the IEEE SMC Information Assurance and Security Workshop*, pp. 247–254, 2007.
- [44] N. Svendsen and S. Wolthusen, Connectivity models of interdependencies in mixed-type critical infrastructure networks, *Information Security Technical Report*, vol. 12(1), pp. 44–55, 2007.
- [45] J. Talsma, B. Becker, Q. Gao and E. Ruijgh, Coupling of multiple channel flow models with OpenMI, *Proceedings of the Tenth International Conference on Hydroinformatics*, 2012.
- [46] The White House, Presidential Policy Directive – Critical Infrastructure Security and Resilience (PPD-21), Washington, DC, 2013.
- [47] A. Usov, C. Beyel, E. Rome, U. Beyer, E. Castorini, P. Palazzari and A. Tofani, The DIESIS approach to semantically interoperable federated critical infrastructure simulation, *Proceedings of the Second International Conference on Advances in System Simulation*, pp. 121–128, 2010.
- [48] I. Utne, P. Hokstad and J. Vatn, A method for risk modeling of interdependencies in critical infrastructures, *Reliability Engineering and System Safety*, vol. 96(6), pp. 671–678, 2011.
- [49] M. van Eeten, A. Nieuwenhuijs, E. Luijff, M. Klaver and E. Cruz, The state and the threat of cascading failure across critical infrastructures: The implications of empirical evidence from media incident reports, *Public Administration*, vol. 89(2), pp. 381–400, 2011.
- [50] W. van Leekwijck and E. Kerre, Defuzzification: Criteria and classification, *Fuzzy Sets and Systems*, vol. 108(2), pp. 159–178, 1999.
- [51] A. Vespignani, Complex networks: The fragility of interdependency, *Nature*, vol. 464, pp. 984–985, April 15, 2010.
- [52] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen and D. Wilkins, A comparison of a graph database and a relational database: A data provenance perspective, *Proceedings of the Forty-Eighth Annual Southeast Regional Conference*, article no. 42, 2010.
- [53] S. Walsh, S. Cherry and L. Roybal, Critical infrastructure modeling: An approach to characterizing interdependencies of complex networks and control systems, *Proceedings of the Second Conference on Human System Interactions*, pp. 637–641, 2009.
- [54] D. Zhou, A. Bashan, R. Cohen, Y. Berezin, N. Shnerb and S. Havlin, Simultaneous first- and second-order percolation transitions in interdependent networks, *Physical Review E*, vol. 90(1), article no. 012803, 2014.
- [55] E. Zio and G. Sansavini, Modeling interdependent network systems for identifying cascade-safe operating margins, *IEEE Transactions on Reliability*, vol. 60(1), pp. 94–101, 2011.