URBAN-NET: A Network-based Infrastructure Monitoring and Analysis System for Emergency Management and Public Safety

Sangkeun Lee, Liangzhe Chen**, Sisi Duan*, Supriya Chinthavali*, Mallikarjun Shankar*, B. Aditya Prakash***
**Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA*
{lees4,chinthavalis,duans,shankarm}@ornl.gov
***Department of Computer Science, Virginia Tech, VA 24061, USA*
{liangzhe,badityap}@cs.vt.edu

***Abstract*—Critical Infrastructures (CIs) such as energy, water, and transportation are complex networks that are crucial for sustaining day-to-day commodity flows vital to national security, economic stability, and public safety. The nature of these CIs is such that failures caused by an extreme weather event or a man-made incident can trigger widespread cascading failures, sending ripple effects at regional or even national scales. To minimize such effects, it is critical for emergency responders to identify existing or potential vulnerabilities within CIs during such stressor events in a systematic and quantifiable manner and take appropriate mitigating actions. We present here a novel critical infrastructure monitoring and analysis system named URBAN-NET. The system includes a software stack and tools for monitoring CIs, pre-processing data, interconnecting multiple CI datasets as a heterogeneous network, identifying vulnerabilities through graph-based topological analysis, and predicting consequences based on "what-if" simulations along with visualization. As a proof-of-concept, we present several case studies to show the capabilities of our system. We also discuss remaining challenges and future work.**

***Keywords*-critical infrastructure; network; graph theory; simulation; vulnerability; national-scale**

I. Introduction

Critical Infrastructures (CIs) such as energy, water, transportation and communication are lifeline systems which are vital for public safety and security. CIs are mutually dependent in complex ways and understanding these interdependencies is critical for emergency preparedness, sustainability, and reliability. For instance, the energy-water nexus [1] highlights the interdependencies between water and energy systems, where the energy network depends on the water network for energy production, and the water network depends on the energy network for treatment, dissemination, and disposition. For instance, since 2004, water stress within certain regions in the US has led to power plants to temporarily reduce their power output or shut down entirely, and prompted at least eight states to deny new plant proposals [2]. In fact, such dependencies exist across multiple CIs, and make them highly vulnerable where hazards affecting one CI network can potentially propagate to other infrastructure networks and disrupt the functionality of the entire system.

The very recent 2016 Puerto Rico blackout [3] is an example that shows that a disruption of a substation affects several CIs simultaneously. Power outages affected nearly 1.5 million customers of the US territory and then caused problems in several infrastructure sectors such as traffic jams, business closures, and water service shutoffs affecting 340,000 people. Several other significant events in recent history such as the 2003 North American blackout, 2012 Hurricane Sandy, and the 2015 Nepal earthquake present examples highlighting the catastrophic effects of these interdependencies.

In order to minimize the negative impact to national security and public safety caused by serious CI disruptions, it is crucial to have certain key capabilities for identifying both existing and potential vulnerabilities in the CIs in a systematical and quantifiable manner. We argue that the following steps are essential in achieving such a goal. First of all, we need to be able to monitor live extreme events (such as hurricanes, earthquakes, etc.) that can cause dire cascading consequences with subtle but important differences in impact for each given event. Second, it is necessary to robustly model various kinds of massive CI networks and their operations so that we can rapidly identify existing vulnerabilities. Finally, predicting potential vulnerable components and damage under certain scenarios (e.g., predicted or actual hurricane path) is a necessary requirement to enable appropriate mitigative actions.

However, realizing a system with such functionality is not straightforward because we need to consider several layers of multidimensional (such as types of interdependencies, types of failures, coupling, etc.) data at various scales of time and space. The breadth and complexity of large-scale CI data further exacerbates the problem. A number of previous

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

efforts addressing critical infrastructure dependencies [4], [5], [6] have been limited by their considering a limited set of critical infrastructure and performing qualitative analysis as opposed to quantitative analyses. To the best of our knowledge, no existing system provides a software stack that performs vulnerability analysis for national-scale interdependent CI networks.

In this paper, we present the current status of our URBAN-NET system, which aims to provide monitoring for the CIs and vulnerability analysis for national-scale heterogeneous interdependent CI networks. URBAN-NET system has been designed to include a set of network oriented tools that can be a part of a CI monitoring and analysis work-flow. Specifically, URBAN-NET comes with a suite of tools for 1) pre-processing the geospatial physical infrastructure datasets, 2) monitoring the status of CI networks, 3) flexibly building a network of CI networks (i.e., a heterogeneous CI network), and 4) performing both graph topological-based and simulation-based algorithms with complex networks. We also visualize the results to provide relevant insights for a decision maker. We build on various Big Data, GIS (Geographical Information System), and graph-theoretic technologies for development of each component.

The rest of the paper is organized as follows. In Section 2, we introduce background and related work. Section 3 presents an overview of URBAN-NET system focusing on the generic work-flow and objective of each component. In Section 4, we describe the detailed implementation of URBAN-NET along with our best practices and case studies. Finally, we conclude the paper in Section 5.

II. Related Work

A. Infrastructure Vulnerability Analysis

Modeling and analyzing vulnerability of interdependencies within critical infrastructure networks are challenging tasks, since it involves multiple dimensions such as the types of coupling and failures and type of interdependencies [7], [8]. For instance, there exists a physical interdependency between power substations and oil refineries for energy generation, and between water stream network and generation power plants for water cooling. During a natural disaster such as an earthquake, there exists geographical interdependencies across all the CI networks. Also, there exist cyber interdependencies between control networks and the corresponding CI networks where a cyber attack can cause severe business and infrastructure losses. damage [9], [10], [11].

A number of previous works focus on analysis and simulation of CI networks, which are broadly categorized into: empirical approach, agent based, system dynamics based, economic theory based, and network based approaches [12]. There exist a number of mathematical frameworks [13], [14], [15] and interdependency models [16], [17], [18] for understanding the robustness of interdependent networks. However, most of them focus on at most two CIs, which limits their practicality. Furthermore, only a limited number of previous efforts in interdependency analysis focus on the the impact of natural disasters, which limits their effectiveness in emergency management. ORNL's EARSS models enables the analysis of weather related threats such as hurricanes to the CI networks [19]. However, its usage is still limited to two CI networks. In comparison, the goal of our system is to model multiple CI networks of large geographical region/network locations based on real-world datasets.

B. Infrastructure Network Monitoring

A number of existing works utilize both GIS tools and real-time natural disaster datasets for hazard analysis [20], [21], [22]. Most of them focus on the analysis of a single network. For instance, landslide hazard analysis can be performed based on real-time data such as SPOT images [21] or remote sensor data [22]. On the other hand, a framework for both infrastructure monitoring and management was proposed in [23]. However, it does not consider the interdependencies between networks and therefore lacks the practicality in emergency management. ORNL developed tools such as VERDE provide a wide-area situational awareness of the grid infrastructure by allowing a decision maker to overlay live weather, population data and distribution outages to visualize the health, criticality and risk of utility [24]. In this work, We inherited several features of these tools for URBAN-NET's network monitoring functionality. Indeed, due to the development of technologies in infrastructure network monitoring [25], [26], the health of infrastructures can be reported in near real-time. However, we argue that data analysis is the key to emergency management. Our approach collects data from various utility companies and social media for reports such as natural disasters and outages. Combined with the interdependency analysis, we will be able to provide a national-scale holistic view of the infrastructure networks and a comprehensive capability for efficient emergency management.

C. Big Data and GIS

There are a number of big data tools for large scale data analysis, such as MapReduce [27] and Spark [28]. Both tools provide scalability and fault-tolerance for applications written in map-reduce programming model to processing of large-scale datasets. To leverage advantages of such big data tools, URBAN-NET's data processing tools are mainly written following the map-reduce programming paradigm. Infrastructure analysis desires the capability of handling geographical data. Geographical Information Systems (GIS) provide the capability to efficiently store, retrieve, analyze, and visualize spatial data. In order to support a GIS, several geographical DataBase Management Systems (DBMS) can be employed. We utilized PostGIS [29], which is an open source extension that supports geographical data to the PostgreSQL DBMS. In order to support the processing large-scale geographical datasets, previous approaches proposed frameworks above MapReduce that support GIS features [30], [31]. Our previous work [32] proposed an Infrastructure Vulnerability Analysis System that utilized both big data tools and GIS tools for data management and data analysis. In this work, we further extend our previous framework and support other functionality such as infrastructure monitoring. As a result, our system can be useful in both data analysis and emergency management.

D. Network Analysis Algorithms and Tools

The graph representation of CIs has an advantage since it can provide intuitive visualization to CI operators. Various data formats are available. Shapefile [33] is one of the very popular geospatial data format for GIS tools. The vector features of Shapefiles, such as points, lines, and polygons, can perfectly represent CIs, e.g., rivers, roads, and substation service areas. We utilized Shapefiles for CI networks identities and visualization tools such as OpenLayers [34]. For network analysis, we convert them to other data formats such as CSV files [35] and plaintext and utilize Big Data tools to achieve better performance.

CI operators are often interested in quantifying the vulnerability of graph components (e.g., nodes, edges, subgraphs, etc.), since it can provide more specific information to CI operators. A few previous efforts incorporate simple graph-theoretic measures such as node degree or betweeness [36] for vulnerability analysis [37], [38]. For computation of the graph measures, existing tools such as gm-sparql [39] or NetworkX [40] can be used for computing various graph measures such as *node degree*, connected component [41], and node eccentricity [42] within a triple store.

Note that most of the conventional graph measures are defined on a homogeneous graph, which is composed of a single type of node and edge. In comparison, CI network-of-networks graphs are heterogeneous. Therefore, for analysis of CI networks, instead of directly using such measures, we need to define new measures for identifying vulnerabilities in CI networks incorporating domain knowledge of different CI networks. We aim to leverage combinations of existing graph-theoretic measures for computing vulnerability measures and processing networks, including *node degree*, *shortest paths*, connected component [41], etc.

III. Overview of URBAN-NET

The goal of URBAN-NET system is to provide a front-end that can be used by subject matter experts to support their decision making based on the vulnerability of the CI infrastructures. In order to achieve such a goal, there are many crucial back-end tasks such as data processing (e.g., conversion of file formats, constructing networks, etc.) and data analytic at large-scale. Accordingly, the URBAN-NET system not only refers to the front-end user-interface with visualizations but also refers to a full-stack of back-end software and hardware systems including data processing, analytic algorithms and their implementations. Figure 1 shows the conceptual architecture of URBAN-NET system and how its components of the system can interact with each other.

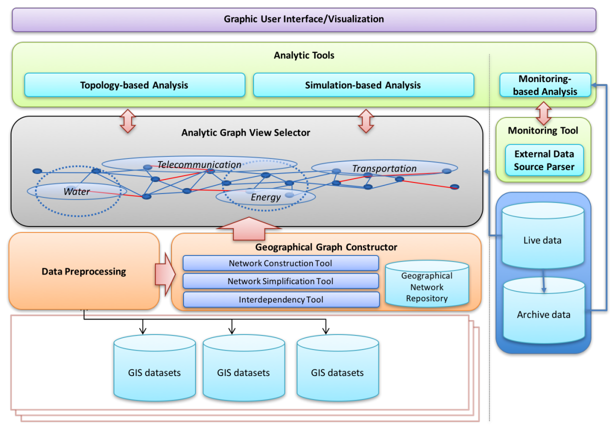


Figure 1: Conceptual architecture of URBAN-NET system

Data processing: There exists a wide range of CI datasets compiled by federal agencies or commercial vendors. For instance, the United State's National Geospatial-Intelligence Agency (NGA) and the Department of Homeland Security (DHS) published a unified infrastructure geospatial data inventory, namely HSIP Gold [43], which includes domestic infrastructure datasets collected from various government agencies and partners. NHDPlus [44] is a dataset created by the US Environmental Protection Agency (EPA), which includes information about the nation's hydrological framework. Other examples include open-source Energy datasets from U.S. Energy Information Administration (EIA) [45]. We notice that majority of such geographical these datasets are being published in the *Shapefile* [33] format and commonly used in majority of GIS systems. Thus, URBAN-NET is designed to use *Shapefile* as base datasets for analysis. However, despite of *Shapefile* format's advantages such as interoperability and prevalence, it is not directly usable for URBAN-NET's analysis, as the format is rather focused on visualizing vector components on a map. In comparison, URBAN-NET's analytic tools take graph-theoretic approaches to analyze the CI networks while Shapefiles are not designed for performing network-based analysis. More importantly, it is also difficult to utilize relationships of entities in different Shapefiles.

Thus, the lowest layer of URBAN-NET is responsible for transforming Shapefile datasets into the network (i.e., graph) datasets that are more analysis-friendly, where for each component its adjacent or linked components can be identified. URBAN-NET converts Shapefile datasets into network datasets in the following order. First, it transforms Shapefiles into CSV (Comma Separated Values) files. Note that most big data tools such as MapReduce frameworks are designed to use text files that can be easily split as its input data, so the first step is transforming Shapefile, which is in binary data format, into flat text file. Second, using a CSV file as its input, it generates vertices and edges. Not all Shapefile data will generate both vertices and edges; for example, if the original data set only includes information about substations and their locations, the generated network will only contain vertices; as relationship or linkages information is not in the original file. Third, if needed, URBAN-NET simplifies the outcome of the second step. Some networks constructed from a Shapefile can be overly complicated and may include lots of vertices and edges that are not really useful for network-based algorithms. The third task aims to reduce the number of nodes and edges to make the network more meaningful and processible. Initially, network constructed from different datasets are independent and not connected to each other. The last step is to create interdependency links across different networks, which will integrate multiple different datasets into one network.

The constructed network datasets are stored in the network repository; then subset of the constructed can be composed as a view for specific analytic tasks. It is important acknowledge that data processing itself involves with many research questions and implementation challenges in terms of scalability. We will discuss the details of each step and related tools in Section IV.A.

Network monitoring: To monitor status of CI networks such as the electrical grid, the network monitoring tool collects data from various data sources, parses and stores them into the network status database. The goal of network monitoring can be summarized into three major objectives. First, it aims to expose dynamic status of a CI network to subject matter experts so that they can be aware of a emergency situation in near-real time. Second, it allows to archive historical status of CI networks so that they can be later used as base dataset for predictive or statistical analysis. Last, it aims to use real-time event as a trigger of a simulation-based analysis's input so that the system can automatically initiate performing useful analysis without having to be requested by subject matter experts. The current URBAN-NET system focuses mainly on the electrical grid and outage status as its example implementation, inheriting features from ORNL developed tool VERDE/EARSS [24]. However, the concept can be extended to other domains such as gas pipe line, etc. We discuss the details such as data sources, data collection and storage, coverage of current implementation, etc in Section IV.B.

Network-based analysis: In the higher layer of URBAN-NET, we provide network-based analysis to answer these questions: (1) how efficient is the CI network as a whole? What are the most important nodes/edges in the network whose removal cause largest impact? (2) What are the consequences of a given event (like a hurricane, earthquake, etc.), which triggers a set of initial failures?

The *topology-based analysis* aims to answer the first question. The challenges here are how to characterize the efficiency of the network with different types of nodes and edges, and how to quantitatively measure the importance of nodes and edges. The current version of URBAN-NET provides two quantitative measures for CI network analysis that are *efficiency score* and *reachability score* for network components. In this work, we show examples of topology-based analysis using the road-gas station network.

The *simulation-based analysis* answers the second question. It aims to understand how effects of perturbation events in an infrastructure (e.g., damage in the road network) can spread out across multiple infrastructure networks via simulation. Directly adopting existing propagation models such as random-walks [46] and label propagation [47] may not be appropriate for this task. These methods usually assume local propagation where failure/labels propagates through links, while in CI networks, a failure of a node/edge can cause failure in disconnected nodes/edges. Hence, we propose a rule-based simulation for this tasks, which captures the main physical interdependencies between nodes/edges. Our simulation with various what-if scenarios such as random perturbations (inactivating random nodes or edges), targeted perturbations (inactivating nodes with higher number of edges), regional perturbations (inactivating a set of nodes or edges located in a region) can be useful for CI operators. Further, we provide visualization of our simulation results to support better decision making.

IV. URBAN-NET Tools and Best Practices

A. Data processing tools

URBAN-NET's data processing tools provide methods for users to flexibly build network datasets using given Shapefile datasets, depending on the analysis that needs to be performed. A *network dataset* refers to a dataset where its data components are composed of a set of vertices (i.e., nodes) and a set of edges. Generally speaking, a vertex represents an entity on the map, which usually contains its geographical location and other properties. For instance, a substation can be represented as a vertex in the network dataset. On the other hand, an edge represents a relationship between entities; for instance, transmission line connecting from a substation to another substation can be represented as an edge. In some cases, a vertex or an edge does not represent physical entities, and they only represent geographical locations. For example, in case of road network, an edge representing a road (or a part of road) is a linkage between two nodes representing geographical locations. We use node list and edge list file to represent a network (See an example in Figure. 2).

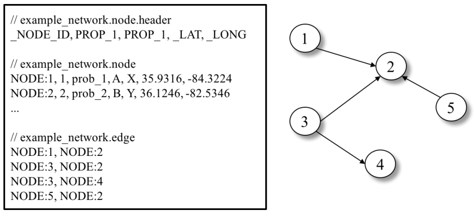


Figure 2: Example of node list, edge list, and header files

Network data construction is done by performing a series of tasks. Figure 3 shows the pipeline of using data processing tools to create a CI networks. **shp2csv** transforms *shapefile* format datasets, which are binary datasets that are not distributed processing friendly, into flat CSV files that are more preferable by most Big Data tools. Internally, it relies on *shp2pgsql* tool that is included in the PostGIS software package, which imports the shapefiles into a PostGIS/PostgreSQL database table. After importing a shapefile dataset into a temporary database table, *shp2csv* exports the table to a CSV file and creates a header file that describes the columns of the table. Separating header from data allows easily splitting data into multiple chunks and processing them in parallel manner.

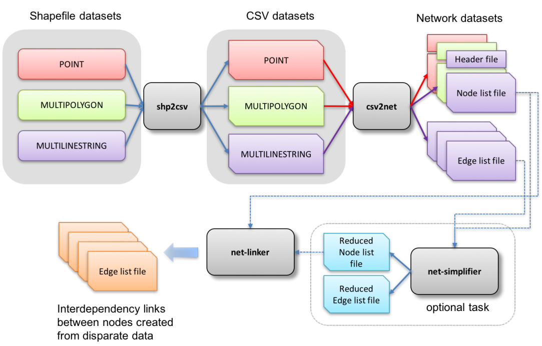
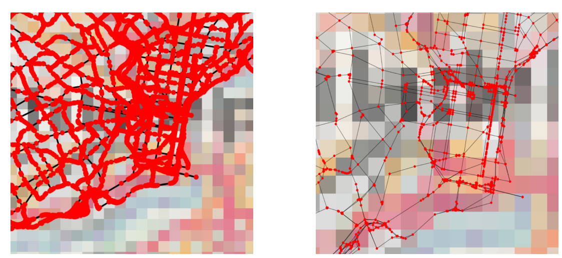


Figure 3: Data processing pipeline

The current version of *shp2csv* supports POINT, MULTILINESTRING, and MULTIPOLYGON types of shapefile. To be more specific, a POINT type shapefile is composed of a set of geographical locations and their properties. A MULTILINESTRING shapefile includes entities where each entity represents a set of lines (i.e., series of geographical locations, e.g., river stream or street). A MULTIPOLYGON shapefile contains a set of entities that represent a set of polygons (i.e., closed loop of geographical locations, e.g., service area).

Next, **csv2net** tool takes a CSV file generated by *shp2csv* as its input and converts the file into a node list file, a node header file, and an edge list file. It processes every row in the input CSV file and generates nodes and edges accordingly. If a row represents a POINT or a MULTIPOLYGON object, it is transformed into a vertex. A unique value is then assigned to the node as a identifier _NODE_ID, and the node inherits property values of the object. In case of POINT, the _LAT (latitude) and _LON (longitude) values are extracted from the original object and added to the constructed node property value. Similarly, _GEO property whose value is a sequence of (_LAT, _LON) are added to the node constructed from MULTIPOLYGON as a property value. If a row represents a MULTILINESTRING, each geographical location that composes a line are converted to a node, then the connections between nodes are converted into edges. Similar to the nodes generated from POINT shapefile, a node identifier is assigned and properties are inherited to each node. This conversion process is embarrassingly parallel and implemented following the map-reduce programming model for scalable processing. More specifically, map function takes care of the node and edge generation logic, no reduce task is required. We implemented the tool in Python, and it runs on Apache Hadoop [48] via Hadoop streaming [49].

net-simplifier performs network simplification, which is an optional task, but very useful in most cases. Because shapefiles are originally developed to precisely describe a shape of an object, in many cases, too many vertices and edges can be generated if we directly convert them into a network. Thic can be problematic because too many vertices and edges can negatively affect not only performance in terms of processing time but also the accuracy of analysis. *net-simplifier* filters out nodes whose edge degree equals 2, then directly connects two nodes connected via the removed node, so that the terminal nodes and intersection nodes remain. Figure 4 shows how a complicated road network constructed from a MULTILINESTRING shapefile can be simplified the tool.



(a) Before simplification (b) After simplification

Figure 4: Comparison of road network before and after simplification

Similar to *csv2net*, we implemented this functionality based on a map-reduce programming, and it is processed in 4 steps, as shown in Algorithm 1. The first map-reduce phase computes edge degree for each node from the given edge. For every node in the given edge list E=$\{e_1$,e_2,$\dots\}$, map function emits (start_node_ID(e),1) and (end_node_ID(e),1), then reduce function aggregates the values by keys. In the second phase, vertices are filtered using a map function by checking if the degree of each vertex does not equal to 2.

Algorithm 1: Pseudo code of map function to create simplified network

```
Data: linestring list L={l_1,l_2,...}
  function MAP(l)
    // Getting the sequence of vertices {v_1,v_2,...}
    v_prev←φ;
    V ← getVertexList(l);
    // Create edges to keep and emit them
    For v_i in V
    if v_prev=φ then
      if checkToKeep(v_i) then
          v_prev← v_i;
      // checkToKeep(v_i) returns True,
      // if degree(v_i)≠2 or,
      // v_i is the first or the last vertex of l
    else
      if checkToKeep(v_i) then
          emit(e(v_prev,v_i));
          // Emitting an edge v_prev→v_i
          v_prev←φ;
      else
          pass;
```

The third phase is to create vertex sequence of original shape of MULTILINESTRING. The map function of this phase takes the edge list file and emits the object IDs of the original shapefile objects as key along with other edge property values, then the reduce function aggregates them and reconstruct linestring sequences of vertices. The output result can be formalized as a set of linestrings L={l_1, l_2, ...}, where $e_{i,j}$ represents jth element of linestring l_i. The final phase reproduces edges while checking the degree of nodes at each side of the edge using a map function.

Network datasets created by the previous tools create a network from a single data sources. **net-linker** performs the last task in the data processing pipeline, which is to create interdependency links across nodes constructed from disparate datasets. The current version of URBAN-NET, we mainly focus on geographical interdependencies. We support two rules to create edges between nodes that are *nearest-neighbor* and *contained-in-polygon*. The tool takes two network datasets as input data. For every node in a network data set, the *nearest-neighbor* rule identifies the nearest node in the other given network and generates edges between them. Similarly, *contained-in-polygon* creates edges if a node in a network is geographically contained in a geographical shape of a node. Note that *contained-in-polygon* is applicable only if one input network contains nodes constructed from MULTIPOLYGON shapefile objects. Performing geographical operations such as *finding nearest points*, *determining overlaps*, and *measuring distance* without proper indexing techniques can be very expensive. So we leveraged the indices (e.g., R-Tree-over-GiST [50], etc.) supported by PostGIS to efficiently generate interdependency links in the implementation of *net-linker*.

B. Network monitoring tools

URBAN-NET *network monitoring tools* provides situational awareness of CIs by providing a holistic view of the health of the critical infrastructures. Existing version mainly provides information about the US electric grid while the concept can be applied to all other CIs. These monitoring tools extract near real-time data from various sources such as websites and social media of a number of utility companies across the country by periodically crawling and parsing the outage information. The data is then aggregated and stored in a POSTGIS database for analysis. We mainly provide two tools through the *external data parser* implementation: *National Outage Map Parser* and *Social Media Parser*.

The **national outage map parser** crawls data from the outage websites of around 200 utilities every 15 minutes [51]. It parses the outage information and aggregates outage data at the county level. Specifically, for each county, we provide the number of outages, number of customers affected, total number of customers served, and other information such as the expected restoration time. The data is then visualized as a choropleth map using tools such as OpenLayers [34], as illustrated in Figure 5. Combining this *live data* obtained from *external data parser* with weather data from NOAA, landscan population, hazard layers from USGS, USDA Forest service etc can quickly show the correlations between weather and impacts on the distribution grid.

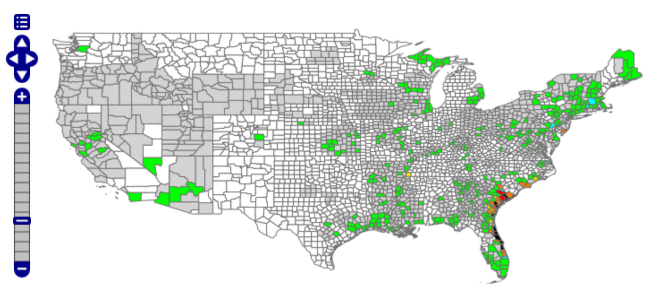


Figure 5: National Outage Map visualization during the Hurricane Matthew

The **social media parser** extracts real-time power outage information from social media streams (e.g. twitter) of several utility companies [51]. Relying on streaming data from twitter provides situational awareness for regions where we have no coverage within the national outage map. This tool crawls tweets published by several twitter accounts of the utility companies, filters unrelated information, aggregates the necessary outage information such as number of outages and location of the outage, and stores it into the database.

The data is then visualized as a choropleth map based on the number of tweets published by various utilities using OpenLayers. The *archived dataset* collected from the *external data parser* supports identification of network recovery patterns within the distribution grid.

C. Analytic tools

In this layer, we provide network-based analytic tools to further analyze the network constructed by URBAN-NET. Our topological-based analytic tool takes the heterogeneous network constructed by URBAN-NET, and outputs the importance score for each node and edge in the network. Our simulation-based analytic tool takes a set of initial failure nodes/edges as input, and predicts how the failure cascades through the network and estimates the final scale and impact of the event.

1) Topological analysis: The goal of this tool is to identify important components and interdependency links, the failures of which may cause catastrophic effect on the entire network. There can be many different quantitative measurements of the node/edge importance depending on the type of CI network we study. In the following, we show as a concrete case study how such a measurement can be designed on a network with two different types of nodes.

In this case study, we create a heterogeneous graph composed of road network and gas stations constructed by URBAN-NET. We use G_R, G_G to denote the two individual networks, and V_R, V_G the set of nodes in the two networks. We interlink the two networks by connecting each gas stations to the closest two road nodes. Then we define the following efficiency score for the entire network:

$$\alpha(G_{RG}) = \underbrace{\frac{1}{|V_R|(|V_R|-1)} \sum_{v_i,v_j \in G_R, v_i \neq v_j} \frac{1}{d_{v_i v_j}}}_{\text{transportation efficiency}} + \lambda \underbrace{\frac{1}{|V_R|} \sum_{v_i \in V_R} \frac{1}{d^*_{v_i}}}_{\text{reachability of gas stations}}$$
$$\text{where } d^*_{v_i} = \min_{v_j \in V_G} d_{v_i v_j} \tag{1}$$

where $d_{v_i v_j}$ is the length (sum of edge weights) of the shortest path from v_i to v_j, $d^*_{v_i}$ is the length of the shortest path from v_i to its closest gas station, and λ represents the weight we put on the second component in Eq. 1. The transportation efficiency component calculates the average inverse of distance between all node pairs in G_R, and the reachability of gas station component measures the average inverse of distance from any node to its closest gas station. Hence, a higher value of $\alpha(G_{RG})$ indicates a better G_{RG} as it is easier to reach other road nodes and the gas stations in the network. Then we can define the importance of a node/edge naturally by calculating the decrease of the overall

Algorithm 2: Importance score $\Delta\alpha$ calculation

Data: G_R, G_G, λ
Result: $\Delta\alpha$ for all nodes and edges.
Construct G_{RG} from G_R, G_G as described in Section IV-C1
SP={}, t=0, r=0//Initialization
for $v_i, v_j \in V_R$, *and* $v_i \neq v_j$ **do**
  SP[v_i][v_j]=shortest path from v_i to v_j
  t+=$\frac{1}{d_{v_i v_j}}$
for $v_i \in V_R$ **do**
  Find the closest $v_j \in V_G$
  SP[v_i][v_j]=shortest path from v_i to v_j
  r+=$\frac{1}{d_{v_i v_j}}$
t=t/$|V_R|(|V_R|-1)$, r=r/$|V_R|$
$\alpha(G_{RG}) = t + \lambda r$
$\Delta\alpha = \{\}$
for *every node/edge x in G_{RG}* **do**
  Construct G'_{RG}
  Recalculate SP[v_i][v_j] if it contains the x
  $\Delta\alpha$(x) $= \frac{\alpha(G_{RG})-\alpha(G'_{RG})}{\alpha(G_{RG})}$

network efficiency upon its removal:

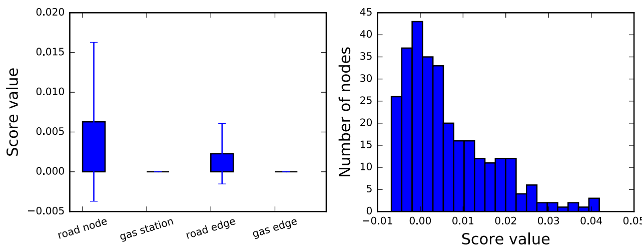
$$\Delta\alpha(v_i) \text{ or } \Delta\alpha(e_i) = \frac{\alpha(G_{RG}) - \alpha(G'_{RG})}{\alpha(G_{RG})}$$

where G'_{RG} is the network after removing v_i/e_i, and we normalize the score by dividing the difference with the original network efficiency. The pseudo-code for calculating the $\Delta\alpha$ is shown in Alg. 2. In practice, we use python to implement the code, and calculate the shortest paths in parallel to speed up the process.

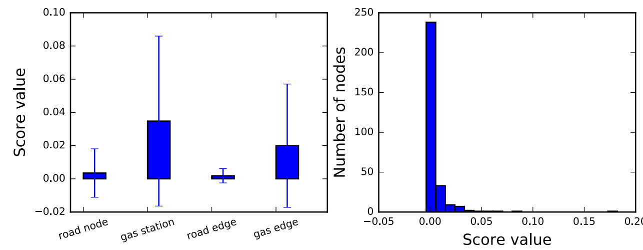
As an example, we visualize our topological analysis results for a small region around Chattanooga in Figure 6. We observe that in both Figure 6(a)(b), we are able to identify important nodes and edges in the network. And in Figure 6(c)(d), we see that road nodes with heavy-traffic, and gas stations close to many road nodes are correctly identified as important. Such analysis can be extended to other types of network as well, and we aim to finally design important score formulation for all major CI networks in the future.

2) Simulation-based analysis: We took a simulation-based approach to allow generation of various what-if scenarios by varying input (such as random perturbations, targeted perturbations, regional perturbations, etc.), and analyze the final output consequences of these input perturbations.

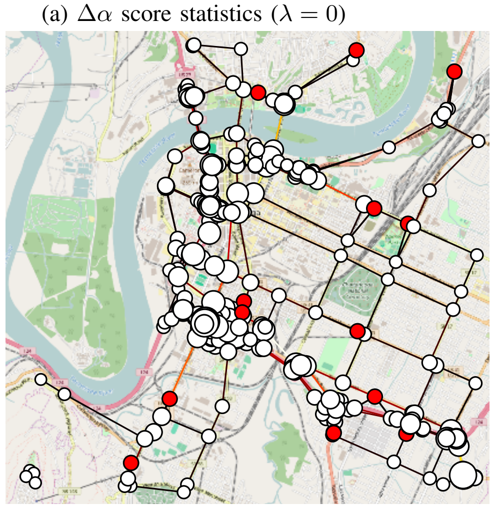
Designing such a simulation system is a very hard task because there are many different types of interdependencies between different CI networks, and it is also hard to identify the temporal factors in the sense that it is not clear when the failure would happen. Also, the timescales of propagation of perturbation is very different for these CIs. While identifying the most important interdependencies between major CI networks and how the failure happens temporally is our final goal, as a first step, we study only the physical



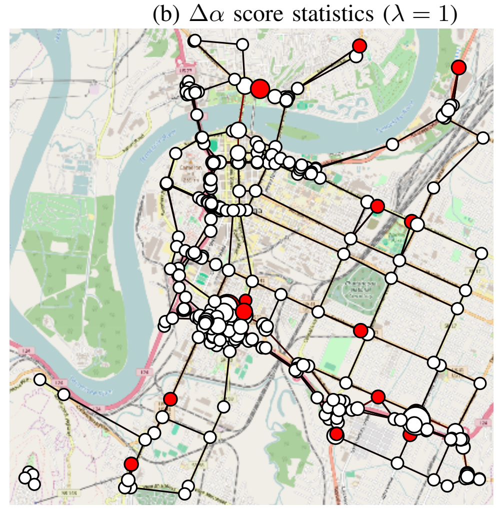
(a) $\Delta\alpha$ score statistics ($\lambda = 0$)



(b) $\Delta\alpha$ score statistics ($\lambda = 1$)



(c) Map visualization for $\Delta\alpha$ scores ($\lambda = 0$)



(d) Map visualization for $\Delta\alpha$ scores ($\lambda = 1$)

Figure 6: $\Delta\alpha$ score results. (a), (b) shows the mean and standard deviation of the $\Delta\alpha(\cdot)$ scores for each type of node and edge. (c), (d) shows the map visualization of the scores, where the node size is proportional to its score, and the color of an edge is also associated with its score using a heat map (the lighter the color, the more important the edge).

interdependencies without considering the temporal aspect. The physical interdependencies refer to the situation where the state of one system is dependent on the material outputs of another system. And we assume that when one node loses its supports, it fails immediately without any resistance (adding resistance and time delay is one of our work in progress).

As an example, we identify important types of components in the power network, natural gas network and transportation network, and use URBAN-NET to construct a corresponding heterogeneous network (see details in Table I). To realize the support chain in the system, we create interlinks between different infrastructures in the following way.

Substations are connected to the nearest transmission bus node since it gets electrical supply from it. Each substation is also connected to the road nodes, natural gas compressors within its service area to capture the fact that it provides power to these local facilities.

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

Infrastructure Type	Node Type	Description
Power	Electrical power plants	Generate electrical power which is transmitted to substations through the transmission network.
	Transmission nodes	Move electrical power from power plants to substations.
	Electrical substations	Transform voltage and distribute electrical powers to consumers.
Natural gas	Natural gas compressors	Increase the pressure of a gas to transport it through pipelines.
	Pipelines	Transport natural gas to consumers.
Transportation	Road intersections & ends	The road network we construct, which contains intersections, end points and their connections.

Table I: Summary for the heterogeneous network constructed by URBAN-NET.

Now we define the following physical-dependency-based rules to simulate the failure cascade:

Substation-Power generator: If a substation does not have a path to any power generator through the transmission network, it fails.
Road-Substation: If a road is not covered by the service area of an active substation, the traffic signals are affected and the road would eventually fail.
Natural gas compressor-Substation: If a natural gas compressor is not covered by the service area of an active substation, it fails.
Power generator (Gas)-Natural gas compressor: If a gas-type power generator (which uses natural gas to generate power) does not have a path to any natural gas compressor through the pipeline network, it fails.

With the above rules, given any perturbation (e.g., inactivate nodes in an area), we can identify the nodes that will fail next, and simulate the results of the final cascade. For example in figure. 7, we construct the heterogeneous network in Florida and visualize them using Gephi, and we randomly choose initial failure nodes at the beginning. We are able to see the how the failure spreads from the initial failed nodes to other nodes. We will develop our own web-based visualization, and our simulation-based tool would eventually be able to estimate the number of affected nodes in each infrastructure network and visualize the foot print of the cascade in an intuitive way for better sense and decision making.

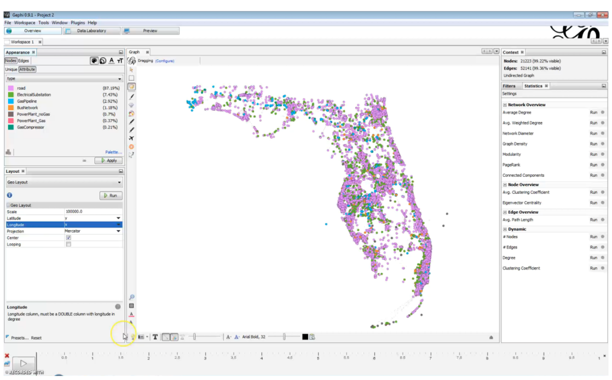


Figure 7: Visualization of our simulation-based tool and results for power, natural gas and road network in Florida.

V. Conclusions

We present URBAN-NET, a critical infrastructure monitoring and analysis system. This system consists of a set of CI network monitoring and analysis tools to provide a holistic view of the health of CIs and their interdependencies. In case of a stressor event such a natural disaster, the tools provide regional and national views of failures based on available data (both raw and processed). The data processing tools transform physical infrastructure data sets and represent these heterogeneous CI networks as network-of-networks for scalable graph-theoretic analysis. With real-world datasets, We demonstrate how URBAN-NET's topology-based and simulation-based analytic tools can enable systematic analysis across CI networks to quantitatively and visually identify existing vulnerabilities within the interconnected networks. For future work, we plan to perform extensive evaluations in terms of scalability and validations of analytic results with real-world scenarios.

References

[1] U.S. Department of Energy, "The Water-Energy Nexus: Challenges and Opportunities," June 2014. [Online]. Available: \url{http://energy.gov/under-secretary-science-and-energy/downloads/water-energy-nexus-challenges-and-opportunities}

[2] Union of Concerned Scientists, "The Energy-Water Collision: 10 Things You Should Know," 2010. [Online]. Available: \url{http://www.ucsusa.org/sites/default/files/legacy/assets/documents/clean_energy/10-Things.pdf}

[3] J. Acevedo, N. Quiñones, and J. Berlinger, "Nearly 1.5 million without power in puerto rico," September 2016. [Online]. Available: http://www.cnn.com/2016/09/21/americas/puerto-rico-power-outage/

[4] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[5] J. Winkler, L. Dueñas-Osorio, R. Stein, and D. Subramanian, "Interface network models for complex urban infrastructure systems," *Journal of Infrastructure Systems*, 2011.

[6] Å. J. Holmgren, "Using graph models to analyze the vulnerability of electric power networks," *Risk analysis*, vol. 26, no. 4, pp. 955–969, 2006.

[7] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *Control Systems, IEEE*, vol. 21, no. 6, pp. 11–25, 2001.

[8] P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: a survey of us and international research," *Idaho National Laboratory*, pp. 1–20, 2006.

[9] M. Abrams and J. Weiss, "Malicious control system cyber security attack case study–maroochy water services, australia," *McLean, VA: The MITRE Corporation*, 2008.

[10] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, 2011.

[11] B. Miller and D. Rowe, "A survey scada of and critical infrastructure incidents," in *Proceedings of the 1st Annual conference on Research in information technology*. ACM, 2012, pp. 51–56.

[12] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability Engineering and System Safety*, vol. 121, pp. 43–60, 2014.

[13] X. Huang, S. Shao, H. Wang, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "The robustness of interdependent clustered networks," *EPFA*, vol. 101, no. 1, 2013.

[14] S. V. Buldyrev, N. W. Shere, and G. A. Cwilich, "Interdependent networks with identical degree of mutually dependent nodes," *Physical Review*, vol. 83, no. 1, 2011.

[15] W. Li, A. Bashan, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Cascading failures in interdependent lattice networks: the critical role of the length of dependency links," *Physical Review Letters*, vol. 108, no. 22, 2011.

[16] M. Parandehgheibi and E. Modiano, "Robustness of interdependent networks: the case of communication networks and the power grid," in *GLOBECOM*, 2013, pp. 2164–2169.

[17] M. F. Habib, M. Tornatore, and B. Mukherjee, "Cascading-failure-resilient interconnection for interdependent power grid-optical networks," in *OFC*, 2015.

[18] S. Duan, S. Lee, S. Chinthavali, and M. Shankar, "Reliable communication models in interdependent critical infrastructure networks," in *Resilience Week (RWS), 2016*. IEEE, 2016, pp. 152–157.

[19] 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*, 2013, pp. 1–3.

[20] D. Fernandez and M. Lutz, "Urban flood hazard zoning in tucumán province, argentina, using gis and multicriteria decision analysis," *Engineering Geology*, vol. 111, no. 1, pp. 90–98, 2010.

[21] C.-T. Lee, "Gis application in landslide hazard analysis–an example from the shihmen reservoir catchment area in northern taiwan," in *Pacific Neighborhood Consortium (PNC) 2008 Annual Meeting program*, 2008.

[22] N. Tassetti, A. Bernardini, and E. S. Malinverni, "Use of remote sensing data and gis technology for assessment of landslide hazards in susa valley, italy," *EARSeL eProceedings*, vol. 7, no. 1, pp. 59–67, 2008.

[23] M. Al-Hader and A. Rodzi, "The smart city infrastructure development & monitoring," *Theoretical and Empirical Researches in Urban Management*, no. 11, p. 87, 2009.

[24] M. Shankar, J. Stovall, A. Sorokine, B. Bhaduri, and T. King, "Visualizing energy resources dynamically on earth," in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*. IEEE, 2008, pp. 1–4.

[25] D. Josephsen, *Building a monitoring infrastructure with Nagios*. Prentice Hall PTR, 2007.

[26] I. Jawhar, N. Mohamed, and K. Shuaib, "A framework for pipeline infrastructure monitoring using wireless sensor networks," in *WTS 2007*. IEEE, 2007, pp. 1–7.

[27] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[28] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," in *HotCloud*, 2010.

[29] R. Obe and L. Hsu, *PostGIS in action*. Manning Publications Co., 2011.

[30] A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz, "Hadoop gis: a high performance spatial data warehousing system over mapreduce," *Proceedings of the VLDB Endowment*, vol. 6, no. 11, pp. 1009–1020, 2013.

[31] Q. Chen, L. Wang, and Z. Shang, "Mrgis: A mapreduce-enabled high performance workflow system for gis," in *eScience, 2008*. IEEE, 2008, pp. 646–651.

[32] S. Lee, S. Chinthavali, S. Duan, and M. Shankar, "Utilizing semantic big data for realizing a national-scale infrastructure vulnerability analysis system," in *SBD@SIGMOD*. ACM, 2016, pp. 3:1–3:6.

[33] E. ESRI, "Shapefile technical description," *An ESRI White Paper*, 1998.

[34] "Openlayers," http://openlayers.org.

[35] Y. Shafranovich, "Common format and mime type for comma-separated values (csv) files," 2005.

[36] L. C. Freeman, S. P. Borgatti, and D. R. White, "Centrality in valued graphs: A measure of betweenness based on network flow," *Social networks*, vol. 13, no. 2, pp. 141–154, 1991.

[37] V. Latora and M. Marchiori, "Vulnerability and protection of infrastructure networks," *Physical Review E*, vol. 71, no. 1, p. 015103, 2005.

[38] L. A. Dueñas-Osorio, "Interdependent response of networked systems to natural hazards and intentional disruptions," Ph.D. dissertation, Citeseer, 2005.

[39] S. Lee, S. R. Sukumar, S. Hong, and S.-H. Lim, "Enabling graph mining in rdf triplestores using sparql for holistic in-situ graph analysis," *Expert Systems with Applications*, vol. 48, pp. 9–25, 2016.

[40] D. A. Schult and P. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, vol. 2008, 2008, pp. 11–16.

[41] L. Qin, J. X. Yu, L. Chang, H. Cheng, C. Zhang, and X. Lin, "Scalable big graph processing in mapreduce," in *SIGMOD*, 2014.

[42] P. Hage and F. Harary, "Eccentricity and centrality in networks," *Social networks*, vol. 17, no. 1, pp. 57–63, 1995.

[43] "Homeland security infrastructure program (HSIP)," https://gii.dhs.gov/HIFLD/hsip-guest.

[44] "NHD plus," http://www.horizon-systems.com/nhdplus/.

[45] "U.S. energy information administration (EIA)," https://www.eia.gov/.

[46] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," 2006.

[47] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Citeseer, Tech. Rep., 2002.

[48] M. Bhandarkar, "Mapreduce programming with apache hadoop," in *Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 1–1.

[49] "Apache Hadoop Streaming," https://hadoop.apache.org/docs/r1.2.1/streaming.html.

[50] T. Nguyen, "Indexing postgis databases and spatial query performance evaluations." *International Journal of Geoinformatics*, vol. 5, no. 3, 2009.

[51] H. M. Supriya Chinthavali, Sankeun Lee and S. Mallikarjun, "Real-time social media data analytics for situational awareness of the electric grid," 2016.