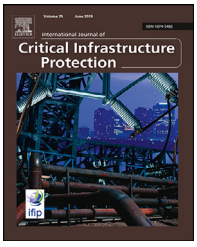
Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/IJCIP



A modeling framework for critical infrastructure and its application in detecting cyber-attacks on a water distribution system

Vishrut Kumar Mishra[a], *Venkata Reddy Palleti*[a,*], *Aditya Mathur*[b]

[a] *iTrust, Center for Research in Cyber-Security, Singapore University of Technology and Design, 487372 Singapore*
[b] *iTrust Center for Research in Cyber-Security, Singapore University of Technology and Design, and Department of Computer Science, Purdue University*

ARTICLE INFO

Article history:
Received 21 October 2018
Revised 18 March 2019
Accepted 7 May 2019
Available online 17 May 2019

Keywords:
Agent-based modeling
Attack detection
Critical infrastructures
Cyber-physical attacks
Cyber-physical systems
Cyber security
Industrial control systems

ABSTRACT

Critical infrastructure (CI), such as systems for water treatment, water distribution, power generation and distribution, is vital for the well being of a society. Such systems are typically large, complex, and interconnected. A cyber-attack on one such system could affect the other. In this work, a generic agent-based framework is proposed to aid in modeling a CI. Combined with a proposed methodology, the framework can be used to model a CI, and those connected to it, as an aid to understanding their collective behavior when subjected to cyber attacks. In a case study, the proposed framework was used to create a model of an operational water distribution system and validated experimentally using a set of cyber attacks launched against an implementation of the model.

1. Introduction

Critical Infrastructures (CIs) are systems that provide essential services to the society. Such systems include water treatment, water distribution, power generation and distribution. Most modern CI consist of a physical plant controlled by a set of Programmable Logic Controllers (PLCs). PLCs obtain plant state via sensors, compute control actions, and send control commands to the actuators to move the plant from one state to the next. The PLCs communicate with each other, and with a Supervisory Control Data Acquisition (SCADA) system, via one or more communication networks. The net effect of plant control is the creation of services or products of value to the economy. Reliability and availability of CI is of paramount importance. However, cyber attacks could have a crippling effect on such CI [1,7,30,50]. While the number of attacks that have crippled or nearly crippled CI remains low, attempts are on the rise and so are the reported vulnerabilities. Thus, it has become necessary to consider ways to protect CI against cyber attacks.

CIs are typically interconnected and mutually interdependent systems. For instance, a water distribution plant requires services provided by power grid, and the power grid may also depend on the services provided by the water distribution plant. The inter-dependencies become more apparent during

* Corresponding author.
E-mail addresses: kumar_mishra@sutd.edu.sg (V.K. Mishra), venkata_palleti@sutd.edu.sg (V.R. Palleti), aditya_mathur@sutd.edu.sg (A. Mathur).

https://doi.org/10.1016/j.ijcip.2019.05.001
1874-5482/

component failures, physical and cyber attacks, and acts of nature [33,43,46]. Therefore, it is necessary to study the characteristics and operations of multiple interconnected CIs through accurate modelling and extensive simulation in the context of cyber attacks and defence mechanisms.

Problem statement, context, and solution strategy: This work addresses the problem of modelling a collection of interconnected CI where each CI is a distributed system. The model will serve as a basis for assessing the effectiveness of a mechanism for detecting cyber attacks against the CI. The following three step strategy is adopted to solve the problem. First, a generic framework (**ABMF**) is created that enables the representation of interconnected and distributed CI (Section 3). Second, a methodology is proposed that uses **ABMF** to create a model from a representation created in the first step (Section 4). Third, the **ABMF** and the methodology is applied to an operational water distribution plant to create a model (Section 5). This step serves to demonstrate the applicability of **ABMF** and the methodology to a realistic distributed CI. Fourth, an invariant-based method [6] is added to the model for detecting any process anomaly in the CI due to cyber attacks. The model, and the added attack detection mechanism, are run in the analysis mode and several attacks are launched to assess how well does the model represent the CI (Section 5.2).

Novelty: Several approaches have been proposed to address modelling CI. These approaches can be categorized as network-based [19,23,38,49], empirical [15,24,27], and agent-based [9,11,14,17,18,20]. Despite the many existing approaches, there is a lack of a generic framework that can be used to model different types of CI and that addresses aspects of multiple CIs in a single framework. Such a generic framework (referred to as **ABMF**) for CIs is proposed in this work. **ABMF** is agent-based where each component of a CI is considered as an agent belonging to one of the following types: commodity, resource, carrier, actuator, and sensor. The agents are linked to one another to create a model of the entire system. Each agent can be reused as many times as required thus making **ABMF** modular that eases modelling a system with multiple similar components. A model so created can be split into any number of sub-models and each sub-model linked to create a complete system model thus making the approach scalable. The model can be used for vulnerability and risk analysis [25,35,40], simulation, behavior analysis, etc.

Contributions: (a) An approach to model critical infrastructures for the purpose of assessing the effectiveness of attack detection mechanisms. (b) A case study using an operational water distribution plant to demonstrate how the proposed approach can be used in practice.

Organization: The remainder of this work is organized as follows. Research related to modelling of CI are listed in Section 2. A framework (**ABMF**) for modelling CI is presented in Section 3. Given a plant design, a methodology to create a model based on **ABMF** is presented in Section 4. The methodology proposed is applied to model an operational water distribution plant (WADI) as a case study. This case study is presented in Section 5. Summary and conclusions from this work are given in Section 6.

2. Related work

Literature review indicates a lack of a generic framework for modeling different types of critical infrastructure. Addressing different aspects of multiple systems in a single framework requires the use of integration. Such a concept for CI is defined in this paper. An agent-based approach is proposed to model components of a CI. Each component, considered as an agent, is categorized into one of five categories: commodity, resource, carrier, actuator, and sensor. These agents are linked to one another to create a model of the entire system. The approach proposed in this paper is contrasted below with those found in literature.

Several reviews summarize models of CIs [21,34,42]. Modeling approaches are classified as empirical, agent-based, system dynamics, and network-based, etc., [34]. The above mentioned approaches have their pros and cons and serve different purposes. None of the approaches is considered the state-of-the-art [22].

Empirical approaches use the historical accident or disaster data to analyze the interdependence of critical infrastructure [4,15,24,27,29]. Such approaches can identify frequent and significant failures and attack patterns. However, they require a large amount of data for accurate modeling. Agent-based approaches use the concept of *emergence* to model complex systems from multiple simpler agents and their interaction. Each component of the system can be modeled as an agent imitating its real-life behavior. The behavior of the system is defined by the interaction between many such agents. Several researchers [11,14,17,18,20] have proposed the use of agent-based models to study the inter-dependencies and analyzed the cascading effects of failures or attacks. Evidently the accuracy of a model depends on the accuracy of the agents that imitate their real-life counterparts.

Existing studies focus on one aspect of the system. For example, [32] describes a detailed input-output model which considers the exchange of different goods and services. Whereas, [48] presents an agent-based Interdependent Critical Infrastructure Model (ICIM) intended to facilitate joint long-term planning. Further, [47] defines the Federated Agent-based Modeling and Simulation which focuses on the systems inter-dependencies. However, there is also a lack of generic model definitions that can be used to model different types of infrastructure with the same agent-related concepts, a goal aimed in the current work. The present work focuses on modeling the physical aspects of a critical infrastructure which can be used to incorporate different methods for anomaly detection when the system is under cyber attack.

Approaches that use system dynamics capture the important cause-effect relationships under disruptive scenarios. These approaches describe the system-level behavior of the infrastructure by using a set of equations [16,26,44] and hence it becomes challenging to analyze component level dynamics. Moreover, such approaches rely heavily on expert inputs. A network-based approach can represent a critical infrastructure by using nodes to represent components and links to represent the connections between the components. [19,23,38,49] uses the topology of the system for modeling

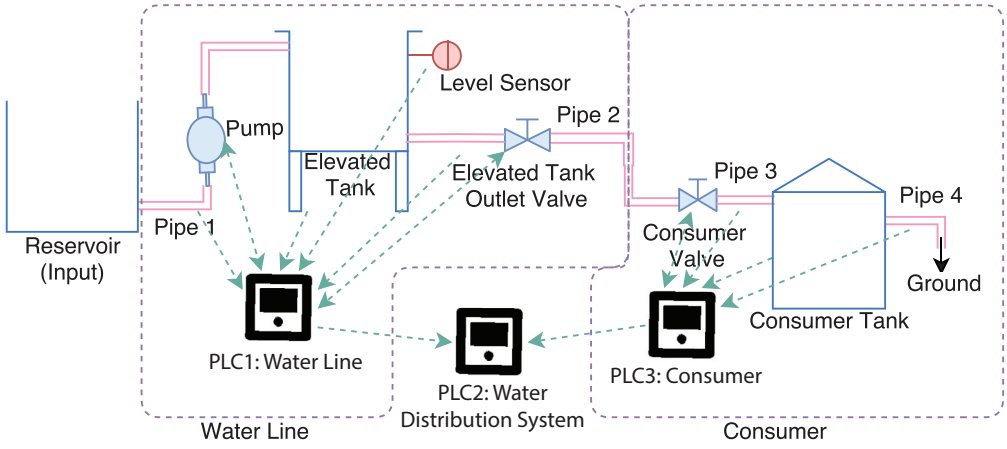


Fig. 1 – A simple water distribution system partitioned into sub-systems "Water Line (WDS-WL)" and "Consumer (WDS-C)".

which makes it easy to perform the topological analysis. Researchers have also focused on networked control security [13,41] and Petri-net based methods [28,40,45,51]. In these works, the discussion is focused on the the vulnerability and risk analysis of critical infrastructures. [35] uses an attack strength degradation model to capture the interdependencies between CIs. Further, [36] studied the impact of potential attacks on critical infrastructures using axiomatic design principles. Recently, [22] provided a unified and convenient framework for modeling the various components that make up a critical infrastructure. The authors use the open hybrid automata framework to model each component.

3. Generic framework for critical infrastructure

This section presents a generic framework for representing an interconnected and distributed CI. A CI is partitioned into multiple sub-systems and components. The proposed framework allows for modeling the entire plant and interconnections–within and across plants. A water distribution system (WDS) in Fig. 1 is used to illustrate the concepts presented in this section. WDS contains a reservoir, an elevated tank, and a consumer tank connected with pipes. As shown, this system can be divided into two sub-systems labelled "Water Line (WDS-WL)" that transports water from the reservoir to the consumer via an elevated tank for storage, and "Consumer (WDS-C)" where the incoming water is consumed. Each subsystem is controlled by two dedicated PLCs, namely PLC1 and PLC3. The overall system is controlled by PLC2. In this section we define the elements of ABMF (Section 3.1) followed by different types of agents (Section 3.2).

3.1. *Elements of ABMF*

Model: A model m is an 8-tuple $\{i_m, I_m, O_m, cd_m, SM_m, A_m, N_m, d\}$, where i_m denotes the model name, I_m the set of inputs, O_m the set of outputs; cd_m the control device in m, A_m the set of agents, SM_m the set of sub-models, N_m the set of nodes, and d the data read from the system, or a third party simulation, or predicted by m; all sets in m are finite.

Agent: A system, or a subsystem, can be divided into one or more components. A component serves as an agent. For example, in WDS-WL agents include the elevated tank, the level sensor installed in the elevated tank, pipe 1 that transports water from reservoir to the elevated tank, pipe 2 that transports water from the elevated tank, the elevated tank outlet valve installed on pipe 2, and the pump attached to pipe 1.

Parameter: Properties of components in a system are referred to as *parameters*. A parameter p is a quadruple $\{n_p, v_p, SSP_p, R_p\}$, where n_p denotes the parameter name, v_p its value, SSP_p is a finite set of set-points, and $SP_i^p = \{l_i^{sp}, fv_i^{sp}\}$, where l_i^{sp} is a label (name) and fv_i^{sp} is the fixed value of the i^{th} set-point of parameter p. For example, in WDS-WL, properties of the elevated tank could be water level, input flow rate, output flow rate, etc. The properties of an elevated tank outlet valve would be its status, i.e., OPEN or CLOSE.

Set point: Each parameter is associated with a value. In practice, such values might not be self-explanatory. To avoid any ambiguity, we define the state of a parameter. Considering WDS-WL let us say the level of an elevated tank can be considered FULL, HIGH, LOW or EMPTY. Similarly, the state of the valve could be OPEN or CLOSE. These states are defined using a set of fixed values. The state could be between two values, greater than or less than a value, or equal to a value. These fixed values define the state of the agent and are referred to as *set-points*. For example, in WDS-WL the state of the elevated tank is considered LOW if the level is less than 30%. Similarly, the state of the pump is OFF if the value is 0.

Condition: A condition is a rule that affects system operation. For example, the opening or closing of a valve can be defined based on the level of an elevated tank. In WDS-WL the rule to define this operation could be "if the water level in the elevated tank is below LOW, the elevated tank outlet valve should be in CLOSE state." Formally, a condition co is a triple $\{n_{co}, r_{co}, de_{co}\}$, where n_{co} denotes the name of the condition, r_{co} is the rule or the condition itself, and de_{co} is the delay, or the reaction time, associated with the condition. The delay represents the time taken by the system to react to a command to move from its current to the commanded state. A condition is categorized as single or double. A single condition must be always satisfied. For example, in WDS-WL the level of the elevated tank should remain above 10%. This condition would

mean that the system is designed such that the water level in the elevated tank never falls below 10%. A level below 10% implies anomaly in the system behavior. A double condition is of the form "A⇒B". For example, in WDS-WL "level of the elevated tank is LOW ⇒ the pump should be ON."

3.2. *Types of agents*

An agent a is a triple $\{i_a, m_a, P_a\}$, where i_a denotes the agent name, m_a its parent model m, and P_a is the set of parameters. Agents are further categorized into six types, namely commodity, resource, carrier, actuator, sensor, and control device. These agent types are described next.

Commodity: A commodity is the product that flows through the system. For example, in WDS-WL the commodity would be water that is flowing through pipe 1, the elevated tank and pipe 2. A commodity *com* is defined as a quadruple $\{i_{com}, m_{com}, a_{com}, P_{com}\}$, where i_{com} and m_{com} are, respectively, unique name and the parent model (m) of the commodity, *com* is in agent a_{com} which could be a resource or a carrier, and $P_{com} = \{P^p_{com}, P^q_{com}\}$ is the set of parameters in the commodity. P^p_{com} is the set of power parameters. It is defined as $\{e_{com}, f_{com}\}$, where e_{com} and f_{com} are, respectively, effort and flow variables. P^q_{com} is a finite set of quality parameters for commodity *com*. A power parameter aids in deciding the flow of the commodity, or the values of parameters, used in simulating the flow. There are two power parameters, namely effort and flow [39]. For example, the power parameter for commodity water would have pressure as the effort and flow rate as the flow variable. The term *power* is defined as the product of *effort* and *flow*. A quality parameter determines the quality of the commodity such as, for example, pH of water or the power factor of the delivered electric power.

Resource: A resource stores or generates a commodity. For example, in WDS-WL the elevated tank is a resource agent. A resource r is a 9-tuple $\{i_r, m_r, com_r, cd_r, P_r, S_r, AC_r, I_r, O_r\}$, where i_r and m_r are, respectively, unique name and parent model, com_r is the commodity, cd_r is the control device to which r is connected, P_r is the set of parameters; the first parameter p^1_r, by default, is *level* of the resource that represents the quantity of commodity present in r, S_r is the set of sensors attached to r, AC_r is the set of actuators attached to r, I_r is the set of inputs; the inputs could be other resources or carriers, and O_r is the set of outputs connected to r. The outputs could be other resources or carriers. Each resource has a parameter that determines the quantity of the commodity present in that resource. For example, "level" of a tank, or "charge" in a battery, are parameters associated with a resource. In special cases, such as for reservoirs, the level parameter is not used as these resources are assumed to have an infinite amount of commodity. Sensors and actuators can be connected to a resource to sense/measure, or control the value of a parameter. Resources also have an array of agents as inputs and outputs which could be any number of carriers or other resources. Most resources are connected to a control device.

Carrier: A carrier connects two or more resources, or carriers, to enable the flow of commodity. For example, in WDS-WL, pipe 1 and pipe 2 are considered as carriers. Analogous to a resource, a carrier c is also a 9-tuple $\{i_c, m_c, com_c, cd_c, P_c, S_c, A_c, I_c, O_c\}$, where each element in the tuple is defined as in a resource. Sensors and actuators can be connected to a carrier to measure, or control, the value of a parameter. Carriers, such as resources, consist of inputs and outputs which could be any number of other carriers or resources. As with a resource, most carriers are connected to a control device.

Actuator: An actuator controls the flow of commodity. For example, in WDS-WL the pump and the elevated tank outlet valve are actuators. An actuator *ac* is defined as a quintuple $\{i_{ac}, m_{ac}, p_{ac}, cd_{ac}, a_{ac}\}$, where i_{ac} and m_{ac} are, respectively, unique name and model name of the actuator, p_{ac} is the parameter the actuator *ac* controls which is usually flow or status, cd_{ac} is the control device to which the actuator is connected, and a_{ac} is the agent to which the actuator is connected which can be a resource or a carrier. Two types of actuators are considered, namely *active* which sets the value of the flow, and *passive*, which can restrict the flow. For example, a pump is an active actuator whereas a valve is a passive actuator. For active actuators the value of output flow rate is the value of the parameter "flow" of the actuator whereas for a passive actuator the value of the output flow rate is the product of the value of the parameter "status" of the actuator and the input flow rate. An actuator is linked to an agent, a resource or a carrier, and controls the value of a parameter of that agent. All actuators are connected to a control device. The control device reads/controls the value of an actuator. For example, in WDS-WL, during simulation the control device will set the elevated tank outlet valve to OFF when the level in the tank is LOW.

Sensor: A sensor is an agent that measures the value of a parameter. It can be observed from WDS-WL that the level sensor is a sensor agent. A sensor s is a 6-tuple $\{i_s, m_s, p_s, cd_s, a_s, E_s\}$, where i_s and m_s denote, respectively, the unique name and parent model, parameter p_s the parameter measured, cd_s the control device to which the sensor is connected, a_s to which the sensor is connected which can be a resource or a carrier, and $E_s = \{ze_{e_s}, se_{e_s}\}$ is a pair of values of error where ze_{e_s} and se_{e_s} denote, respectively, the zero and scaling errors of the sensor. The type of a sensor is defined by the type of the parameter it measures, i.e. power or quality. Zero error is the constant value the sensor outputs when it should output zero; hence the actual value is the value measured by the sensor minus the zero error. By default, the value of zero error is 0. Scaling error is the error in the scaling factor, and is non-zero when the sensor is not scaling the value accurately. Hence, the actual value is the value it measures multiplied by the scaling error. By default, the scaling error is 1. Taking both errors into account, the value measured by a sensor is multiplied by the scaling error and the zero error subtracted from the result. Thus, value = (measured value ∗ scaling error) − zero error

Control device: A control device contains operational conditions that define the next state of the system. Every model has one control device. For example in WDS-WL Programmable Logic Controller (PLC) water line is the control device. A control device *cd* is a quintuple $\{i_{cd}, m_{cd}, A_{cd}, CO_{cd}, CS_{cd}\}$, where i_{cd} and m_{cd} denote, respectively, unique name and model name, A_{cd} is the set of agents connected, CO_{cd} is the set of conditions that the control device uses to define the operation of the system the control device belongs to, and CS_{cd} is the set of constants that the control device *cd* defines. A control device inputs system parameters associated with different agents in the model. Depending on whether it is in simulation or

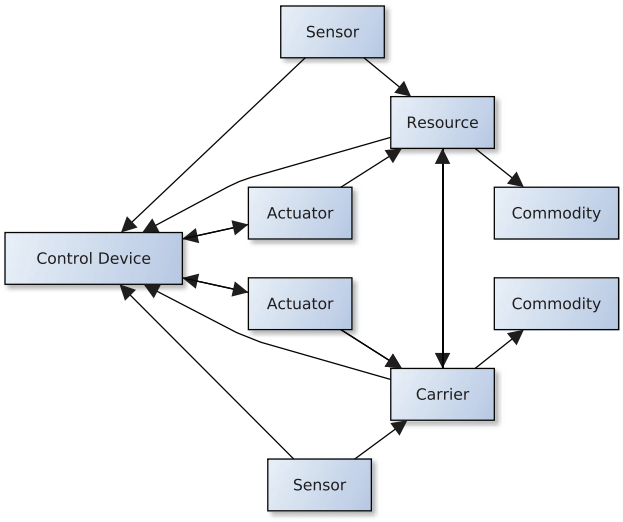


Fig. 2 – Communication between agents.

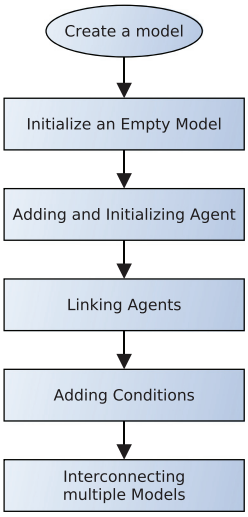


Fig. 3 – Creating a model.

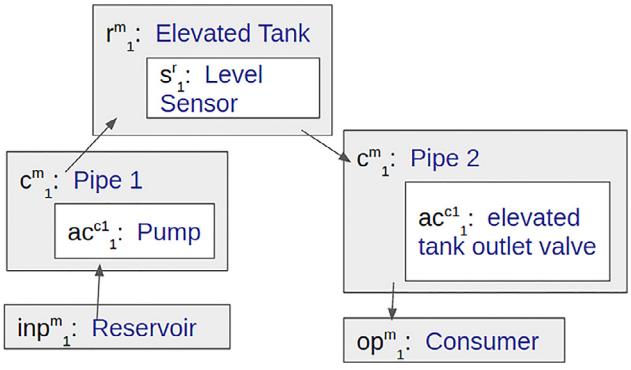


Fig. 4 – Linking agents in the water line subsystem in WDS-WL.

analysis mode, a control device either checks or updates the states of the agents according to the rules. Most agents are connected to a control device. In WDS-WL the control device PLC water line connects to pipe 1, pipe 2, pump, elevated tanks, elevated tank outlet valve and level sensor. A control device contains an array of conditions, or rules, that the model obeys. In the analysis mode, a control device checks each condition to make sure that each parameter is following the associated rules. In simulation mode the control device updates the values/state of the parameters according to the rules. The control device synchronizes different agents in the model according to these rules. The communication i.e. the flow of information between agents is shown in Fig 2. The list of all constants in the model is also saved in the control device. These constants are values that do not vary in a single run but could vary across runs or different experiments.

4. Methodology

Creation and running of a model are described in this section. Steps to create a model is as follows: initialization, adding and initializing agents, linking the agents, addition of conditions, and interconnecting multiple models. Fig 3 shows the steps of creating a model. Steps involved in running the model can vary depending on the objectives. The operation mode of a model is defined based on its objective. Two such modes used in this work are analysis and simulation. In the analysis mode the objective is to check for any model discrepancy whereas in the simulation mode the objective is to predict the state of each agent.

4.1. *Creating a model*

Steps involved in model creation follow. All equations related to the model of the water line system in WDS-WL and WDS-C are included in Table 1 for ease of readability.

Initialization: Creation of model m begins by initialising it with a unique name, inputs, outputs, and control device. The inputs and outputs are resources assumed to have infinite capacity. Model m for the sub-system labelled "Water Line" in Fig. 1 is initialized as in Eq. (17).

$$\begin{aligned} i_m &= water\_line, \\ I_m &= \{In_1^m = reservoir\} \\ O_m &= \{Out_1^m = consumer\}, \\ cd_m &= \{i_{cd} = \text{PLC1}, m_{cd} = water\_line\}, \\ A_m &= \{\}, SM_m = \{\}, N_m = \{\} \end{aligned} \tag{17}$$

Adding and initializing agents: In this step each agent a is initialized with its unique name i_a, the parent model m_a and added to A_m. The resources and carriers in WDS-WL are initialized in Eqs. (4), (7), (9), (11), (13) and (15).

Linking the agents: In this step the sensors and actuators are linked to their respective resources and carriers. Also, as in Fig. 4, resources and carriers are connected to their respective resources and carriers as inputs and outputs.

Sensor s is linked to its respective agent a_s which could be a resource or a carrier. In this example it is the resource r_1^m, i.e. the elevated tank. a_s in Eq. (11) of sensor s_1^m is updated as in

Table 1 – Model and elements of water line system (WDS-WL).

Element	Step 1: Initialize	Step 2: Link
	Model	
Water line(m)		
	$m_0 = \{i_{m0} = simple\_water\_distribution\_system,$ $I_{m0} = \{In_1^{m0} = reservoir\},$ $O_{m0} = \{Out_1^{m0} = ground\}$ $N_{m0} = \{n_1^{m0} = consumer\}$ $cd_{m0} = \text{PLC2}\}$ (1)	$m_1 = \{i_{m1} = water\_line,$ $I_{m1} = \{In_1^{m1} = reservoir\},$ $O_{m1} = \{Out_1^{m1} = consumer\}$ $cd_{m1} = \text{PLC1}\}$ (2)
		$m_2 = \{i_{m2} = consumer,$ $I_{m2} = \{In_1^{m2} = water\_line\},$ $O_{m2} = \{Out_1^{m2} = ground\}$ $cd_{m2} = \text{PLC3}\}$ (3)
	Agent: Resource	
Elevated tank (r_1^m)		
	$i_r = elevated\_tank, m_r = water\_line$ $com_r = \{i_{com}^r = el\_tank\_com, m_{com}^r = water\_line,$ $a_{com}^r = elevated\_tank\},$ $cd_r = \{i_{cd} = \text{PLC1}, m_{cd} = water\_line\},$ $S_r = \{\}, P_r = \{\}, AC_r = \{\}, I_r = \{\}, O_r = \{\}$ (4)	$S_r = \{s_1^r = i_s = level\_sensor\}$ $P_r = \{p_1^r = p_s = level\}$ (5)
		$I_r = \{i_1^r = pipe\_1\},$ $O_r = \{o_1^r = pipe\_2\}$ (6)
Actuator (ac_1^m)		
	$i_{ac1} = pump, m_{ac1} = water\_line$ $p_{ac1} = flow,$ $cd_{ac1} = \{i_{cd} = water\_line\_cd, m_{cd} = water\_line\},$ $a_{ac1} = \varnothing$ (7)	$a_{ac1} = pipe\_1,$ $AC_{c1} = \{ac_1^{c1} = i_{ac1} = pump\}$ (8)
Actuator (ac_2^m)		
	$i_{ac2} = elevated\_tank\_outlet\_vale,$ $m_{ac2} = water\_line, p_{ac2} = status,$ $cd_{ac2} = \{i_{cd} = \text{PLC1}, m_{cd} = water\_line\},$ $a_{ac2} = \varnothing$ (9)	$a_{ac2} = pipe\_2,$ $AC_{c2} = \{ac_1^{c2} = i_{ac2} = ET\_OV\}$ (10)
Sensor (s_1^m)		
	$i_s = level\_sensor, m_s = water\_line$ $p_s = level,$ $cd_s = \{i_{cd} = \text{PLC1}, m_{cd} = water\_line\},$ $a_s = \varnothing, E_s = \{\}$ (11)	$a_s = elevated\_tank$ (12)
	Agent: Carrier	
Pipe (c_1^m)		
	$i_{c1} = pipe\_1, m_{c1} = water\_line$ $com_{c1} = \{i_{com}^{c1} = pipe\_1\_com, m_{com}^{c1} = water\_line,$ $a_{com}^{c1} = pipe\_1\},$ $cd_{c1} = \{i_{cd} = \text{PLC1}, m_{cd} = water\_line\},$ $P_{c1} = \{\}, S_{c1} = \{\}, AC_{c1} = \{\}, I_{c1} = \{\}, O_{c1} = \{\}$ (13)	$I_{c1} = \{i_1^{c1} = reservoir\},$ $O_{c1} = \{o_1^{c1} = elevated\_tank\}$ $P_{c1} = \{p_1^{c1} = p_{ac1} = flow\}$ (14)
Pipe (c_2^m)		
	$i_{c2} = pipe\_2, m_{c2} = water\_line$ $com_{c2} = \{i_{com}^{c2} = pipe\_2\_com, m_{com}^{c2} = water\_line,$ $a_{com}^{c2} = pipe\_2\},$ $cd_{c2} = \{i_{cd} = \text{PLC1}, m_{cd} = water\_line\},$ $P_{c2} = \{\}, S_{c2} = \{\}, AC_{c2} = \{\}, I_{c2} = \{\}, O_{c2} = \{\}$ (15)	$I_{c2} = \{i_1^{c2} = reservoir\},$ $O_{c2} = \{o_1^{c2} = consumer\}$ $P_{c2} = \{p_1^{c2} = p_{ac2} = status\}$ (16)

Table 2 – Attacks launched on WADI.

Attack*	Target(s)	Intention
Single-point attacks on stage 1.		
1.	1_MV_001	Overflow the raw water tank
2.	1_FIT_001	Inject a high dosage of chemicals in the raw water tank
3.	1_AIT_001	Drain raw water tanks
4.	1_AIT_002	Change turbidity set points
Single point attacks on stage 2.		
5.*	2_LT_002	Stealthy attack: Drain the elevated reservoir tank
6.	2_MV_003	Supply contaminated water to the elevated reservoir
7.	2_MCV_007	Cause water to leak from the main pipe line
8.	2_MCV_007	Create an intermittent water supply to consumer tanks
9.*	2_PIC_003	Take control of the booster pump
10.*	2_LT_002	Stealthy attack; overflow elevated tanks
11.	2_MCV_007	Waste water
Multi-point single-stage and multi-stage attacks.		
12.	1_P_005, 1_P_006	Damage a pipe
13.	1_P_001, 1_P_003	Stop dosing of chemicals into the incoming raw water
14.	2_MCV_101, 2_MCV_201	Overflow the consumer tanks
15.*	2_LT_002, 2_LT_001, 1_MV_001	Damage 1_MV_001, the raw water pump, and drain the elevated raw water tank
16.	2_MCV_101, 2_MCV_201, 2_MCV_301, 2_MCV_401, 2_MCV_501, 2_MCV_601	Cut-off water supply to consumer tanks

* Attack not detected.

Eq. (12). Also, S_r and P_r in Eq. (4) of the resource r_1^m are updated as in Eq. (5).

The process of linking an actuator is similar to that of linking a sensor. In this example, ac_1^m is linked to carrier c_1^m and ac_2^m to carrier c_2^m. a_{ac1} and a_{ac2} in Eqs. (7) and (9) of actuators ac_1^m, ac_2^m are updated as in Eqs. (8) and (10). Similarly, S_{c1}, P_{c1}, S_{c2}, P_{c2} in Eqs. (12) and (15) of the carriers c_1^m and c_2^m are updated as in Eqs. (4) and (6). If the actuator is linked to a resource then instead of linking it to (AC_c, P_c), the actuator would be linked to (AC_r, P_r).

Resources and carriers are linked to other resources and carriers as their inputs and outputs. This link is bi-directional. If resource r is linked to carrier c as its input then c is linked to r as its output. Also, the inputs and outputs of the model are connected to their respective carriers or resources. The I_r and O_r in Eq. (4) are updated as in Eq. (6). Similarly, I_{c1}, O_{c1}, I_{c2}, and O_{c2} in Eqs. (13) and (15) are updated as in Eqs. (14) and (16).

Adding conditions: The next step in the modelling process is to link the agents to their respective control devices and update A_{cd} of the control device cd. Set A_{cd} is identical to the set of agents in the model less the commodity. Thus, $A_{cd} = A_m \backslash COM_m$, where COM_m is the set of all commodities in model m.

The next step is to add all conditions CO_{cd} that govern the workings of model m. Each condition is defined by its name, the rule, and the delay. CO_{cd}={co_1^{cd}={n_1^{co}= valve opening, r_1^{co}= if level of elevated tank is less than LOW, then the status of the elevated_tank_outlet_valve should be OFF, del_1^{co}=5}}. Delay is assumed to be 0 for instantaneous action. In case the delay is not known the system is run with no delay and as close to the ideal condition as possible. Then the delay is calculated by taking the maximum of the number of consecutive steps that violate the rule. de_{co} = max number of consecutive steps violating co.

Interconnecting multiple models: A parent model is created to link two or more models or sub-models. The parent model m_0 contains i_{m_0}, I_{m_0}, O_{m_0}, and cd_{m_0}. Each sub-model is added to the parent model m_0 as a set SM_{m_0}. The inputs and outputs of model m_0 are linked to the respective inputs and outputs of each sub-model. Suppose the input $In_1^{m_1}$ of m_1 is to be linked to the input $In_1^{m_0}$ of the parent model m_0. This done by taking $O_{In_1^{m_1}}$, set of all output agents to the input $inp_1^{m_1}$ that is to be linked and adding it to $In_1^{m_0}$ to which the input of the sub-model is linked to. Then, input In_1^{m1} is removed.

The respective input and outputs of each sub-model are linked to each other according to the system design. These links are constructed via nodes $n_i^{m_0}$. The output of one model links to the input of the node and the input of the other model links to the output of the node. Suppose the output $Out_1^{m_1}$ of sub-model m_1 needs to be linked to input $In_1^{m_2}$ of sub-model m_2. To do so, first a node $nd_1^{m_0}$ in the parent model m_0 is created. Then $I_{Out_1^{m_1}}$, the set of all input agents to the output $Out_1^{m_1}$, is added to the set of input agents $I_{n_1^{m_0}}$ of the node $n_1^{m_0}$. Similarly $O_{Out_1^{m_1}}$, the set of all output agents to the input $In_1^{m_2}$, is added to the set of output agents $O_{n_1^{m_0}}$ of the node $n_1^{m_0}$. Afterwards, both input $In_1^{m_2}$ and output $Out_1^{m_1}$ are removed.

Consider system m_0 in Fig. 1 which is created by interlinking models m_1 and m_2 for WDS-WL and WDS-C, respectively. In this case the two sub-models should be linked to each other

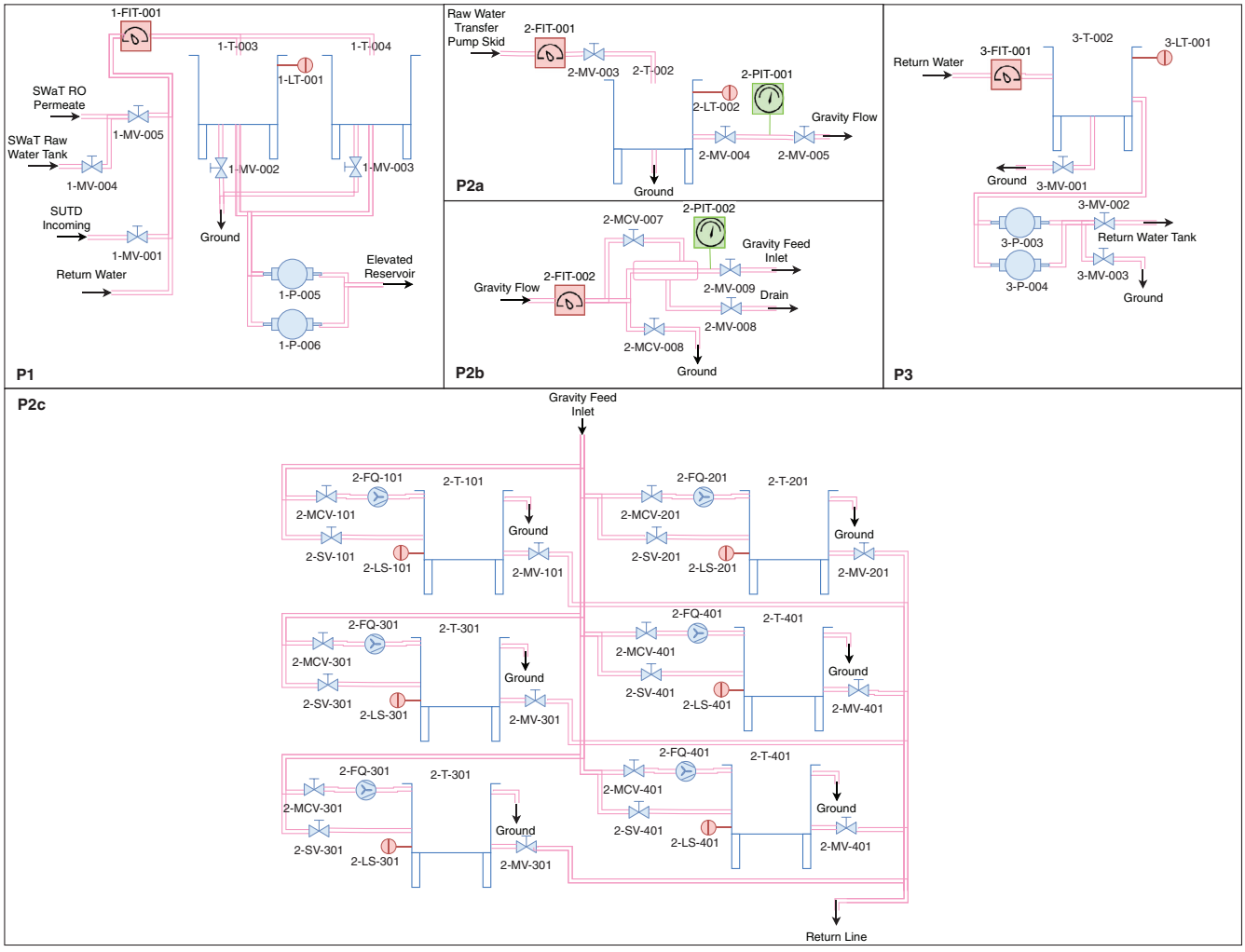


Fig. 5 – Water distribution system representing P1, P2 and P3 stages.

as in Eqs. (2) and (3). The parent model m_0 is initialized as in Eq. (17). Next, the inputs and outputs of the sub-models are linked as follows.

- In_1^{m1} is linked to In_1^{m0}.
- In_1^{m2} is linked to Out_1^{m1} via node n_1^{m0}.
- Out_1^{m2} is linked to Out_1^{m0}.

Note that a node is similar to a sink or a source. Nodes act as an intermediary sink and source to connect two models–sink for one model and source for the other. N_{m_0} is the set of all such nodes of model m_0. The control device cd_{m_0} takes care of all communication between two models. The communication amongst a single model m_i is taken care of by the control device cd_{m_i}.

4.2. *Running a model*

Running of a model depends on the selected mode. The following steps are used when running the model in the analysis mode. (a) Collect plant operational data either from a live plant or saved during a previous run of the plant. (b) Update parameters. (c) Update control outputs. (d) continue with Step (a) if more data available or else stop. After updating the parameters in Step (b), the control device checks all the conditions and any violation is stored in the model. These violations correspond to a process anomaly.

5. Case study

This section describes a case study conducted to assess the applicability of the modeling approach described earlier. The study was conducted in two steps. In the first step an operational water distribution plant (WADI) was selected as the subject and modeled using the methodology presented in Section 4. In the second step an invariant-based attack detection mechanism [6] was implemented and several attacks launched to understand how well the model represents the attacked system.

5.1. *WADI: a water distribution plant*

Water Distribution (WADI) [12] is an operational system used as a testbed for experimentation. It consists of three stages,

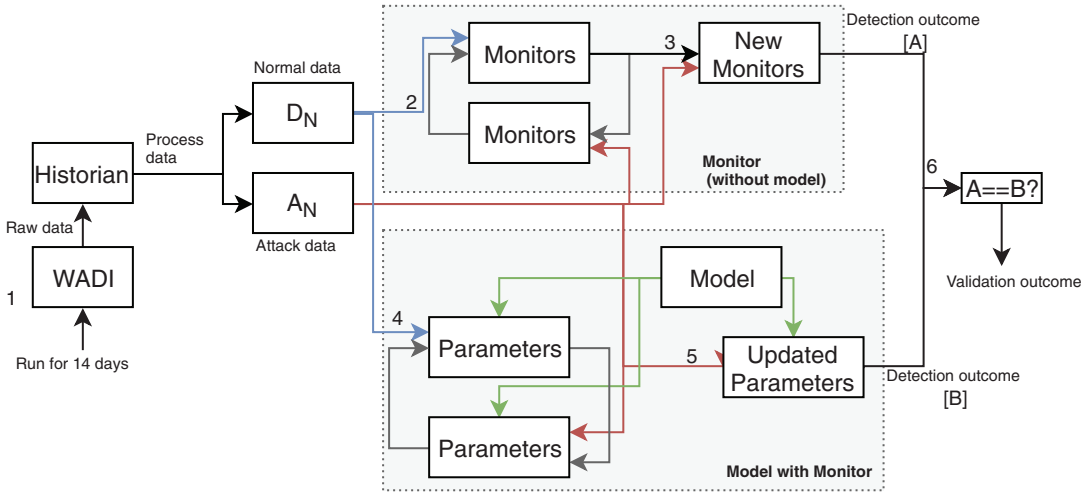


Fig. 6 – Procedure to validate an agent-based model of WADI against process invariants.

namely primary grid (P1), secondary grid (P2), and return water grid (P3). Each stage is controlled by a PLC. WADI consists of 103 sensors and actuators to measure water level, quality, flow rate, pressure, and status of valves and pumps. The sub-models of WADI processes P1, P2, and P3 are in Fig. 5.

P1: The primary grid consists of two raw water (RW) tanks of 2500 L each fed by three incoming sources, namely Public Utility Board (PUB), return water grid, and the Secure Water Treatment (SWaT) plant. Level sensor (1_LT_001) in the primary grid monitors the levels in the RW tanks. Sub-system P1 can be modeled as follows.

$$\begin{aligned} m_{P1} \quad &= \Big\{ i_m = P1, \\ &I_m = \{SWaT\_RO\_Permeate, SWaT\_Raw\_Water\_Tank, \\ &SUTD\_Incoming, Return\_Water\}, \\ &O_m = \{Ground, Elevated\_Reservoir\}, \\ &cd_m = \{i_{cd} = P1\_cd, m_{cd} = P1\}, A_m \Big\} \end{aligned} \tag{18}$$

where the agents in A_m are defined in Tables 3 and 4.[1]

P2: The secondary grid consists of two elevated reservoir (ER) tanks, consumer tanks, and contamination sampling stations. RW tanks supply water to the ER tanks using the raw water pump (1_P_003) installed in the primary grid. Two level sensors, 2_LT_001 and 2_LT_002, are installed in the ER tanks to measure water levels. Water from the ER tanks flows into the consumer tank based on the preset demand pattern. Two water quality monitoring stations are installed at consumer tanks. One station is at the immediate downstream of reservoir and the other is immediately before the consumer tanks (P2A and P2B stations). Once a consumer tank is filled, a level switch raises an alarm and water from the tank drains into the return water grid. For simplicity, the model of P2 is partitioned into three parts, namely P2a, P2b and P2c, defined below.

$$\begin{aligned} m_{P2a} \quad &= \Big\{ i_m = P2a, I_m = \{RawWaterTransferPumpSkid\}, \\ &O_m = \{Ground, GravityFlow, InlineBoosterStation\}, \\ &cd_m = \{i_{cd} = P2a\_cd, m_{cd} = P2a\}, A_m = \{\} \Big\} \end{aligned} \tag{19}$$

$$\begin{aligned} m_{P2b} \quad &= \Big\{ i_m = P2b, \\ &I_m = \{GravityFlow, InlineBoosterStation\}, \\ &O_m = \{Ground, GravityFeedInlet, \\ &\quad BoosterInlet, Drain, Return\}, \\ &cd_m = \{i_{cd} = P2b\_cd, m_{cd} = P2b\}, A_m = \{\} \Big\} \end{aligned} \tag{20}$$

$$\begin{aligned} m_{P2c} \quad &= \Big\{ i_m = P2c, \\ &I_m = \{GravityFeedInlet, BoosterInlet\}, \\ &O_m = \{Ground, ReturnLine\}, \\ &cd_m = \{i_{cd} = P2c\_cd, m_{cd} = P2c\}, A_m = \{\} \Big\} \end{aligned} \tag{21}$$

The agents in A_m for P are in Tables 5 and 6, for P2b in Tables 7 and 8, and for P2c in Tables 9 and 10 respectively.[1] After defining each part, the different parts are interconnected. First, model for P2 is defined as follows.

$$\begin{aligned} m_{P2} \quad &= \Big\{ i_m = P2, SM_m = \{P2a, P2b, P2c\}, \\ &I_m = \{Raw\_Water\_Treans fer\_Pump\_Skid\}, \\ &O_m = \{Ground, Return\_Line, Drain, Return\}, \\ &N_m = \{Gravity, Flow, Inline\_Booster\_Station, \\ &\quad Gravity\_Feed\_Inlet, Booster\_Inlet\} \Big\} \end{aligned} \tag{22}$$

This is followed by linking all the inputs and outputs of the sub-models as defined in the Table 11.[1]

P3: To recycle water, the return water grid pumps water to the primary grid. Water quality analyzers are installed in the return water grid to check water quality before pumping it into

[1] These tables are included in the Appendix.

the primary grid. Model for P3 is defined as follows.

$$\begin{aligned} m_{P3} &= \Big\{ i_m = P3, I_m = \{Return\_Water\}, \\ &\quad O_m = \{Ground, Return\_Water\_Tank\}, \\ &\quad cd_m = \{i_{cd} = P3\_cd, m_{cd} = P3\}, A_m = \{\}\Big\}, \end{aligned} \tag{23}$$

The agents in A_m are defined in Tables 12 and 13.[1]

Interconnection of sub-models: A model m_{wadi} for the entire plant is created by linking the sub-models.

$$\begin{aligned} m_{wadi} &= \Big\{ i_m = Wadi, SM_m = \{P1, P2, P3\}, \\ &\quad I_m = \{SWaT\_RO\_Permeate, \\ &\qquad SWaT\_Raw\_Water\_Tank, SUTD\_Incoming\}, \\ &\quad O_m = \{Ground\}, \\ &\quad N_m = \{ER - RWTPS, R - RL, \\ &\qquad RL - RL, RWT - RW\}\Big\}. \end{aligned} \tag{24}$$

This is followed by linking all the inputs and outputs of the sub-models as in Table 14.[1]

5.2. Attacker model and attack design

A generalized attacker model [3,5] was selected for this study. Three types of attacks were designed using the model as described in [2]: single point attacks on components in one stage, multi-point attacks on one stage, and multi-point attacks across multiple stages. All attacks designed and launched are listed in Table 2.

WADI uses National Instrument's Publish Subscribe Protocol (NI-PSP) in the entire network. We used an attack tool named NiSploit[2] that uses custom LabVIEW Virtual Instruments (VIs) that communicate with shared variables present on different PLCs across the plant using NI-PSP [10]. These shared variables are used by a controller and SCADA to expose data over the network via a shared variable engine. The following two examples from Table 2 illustrate the launch of attacks.

Attack 1: In this attack the intention is to overflow the raw water tank. The attack is launched on the motorized valve 1_MV_001. The related shared variables are in process P1 which contains the current status of the respective motorized valve governing the inflow of water into the raw water tank. The attacker uses this shared variable to execute the attack and change the state of the valve from {1_MV_001=Close} to {1_MV_001 = Open}.

Attack 2: In this attack the intention is to drain the raw water tank and deceive the controller by manipulating the readings of the water quality sensor. The attack is launched on the conductivity sensor 1_AIT_001. Prior to launch, the reading from the conductivity sensor is 176. The attacker manipulates 1_AIT_001 by setting its value to 760 which is different from that received by the controller. As a result the drain valve opens and the raw water tank begins to empty.

[2] https://gitlab.com/gyani/NiSploit

5.3. Invariants and attack detection

An invariant is a condition that must be satisfied in a given state of a system [2,8]. At any given time, monitoring the physical and chemical state of the system against such conditions can act as the basis for detecting process deviations from the expected. For example, the level of water in a tank must always be between its lowest and highest set points; this invariant must always hold during normal system operation. Also, the inlet and outlet flows are dependent on these states. Suppose the inlet valve is open when the level of the tank is at the high set point. Such a state is considered abnormal. In such scenarios, the invariants are useful in detecting attacks on the system that lead to process anomalies.

A total of 35 invariants were derived manually through an examination of the architecture of WADI and the corresponding processes related to the flow of water governed by the pumps. There are 9 invariants for P1, 20 for P2, and 6 for P3. Due to limitations of space below we provide examples of only three invariants, one from each process.

P1:	1_LT_001>70 ⇒ 1_MV_004: Open ∧ 1_MV_001:Close
P2:	2_P_003:ON ⇒ 2_MV_006: ON
P3:	3_FIT_001>0.5 ∧ 3_AIT_005>3 ⇒ 3_P_001:ON ∨
	3_P_002:ON

5.4. Validation procedure

The model of WADI presented above is validated against experimental data. Validation can be performed in at least two ways. One way is to simulate the plant using the model and then compare the behavior exhibited by the simulator with that of the actual plant. The other way, adopted here, is to use the model for detecting cyber attacks and check if these attacks are also detected by the invariants derived from the plant design. The validation procedure consists of steps shown in Fig. 6 and explained below.

Step 1 [Data collection]: The plant was run non-stop in normal mode for 14-days. During days 13 and 14 the plant was subjected to several cyber attacks (see Table 2). All data was saved in the Historian–a server that sits on the plant network and records plant data at regular intervals. Data collected consists of two subsets, namely D_N and A_N. D_N refers to data collected during the first 12-days of normal plant operation while A_N refers to data collected during the last two days. Each data point in the two sets consists of 103 attributes among which 67 are continuous sensor readings and the remaining are actuator states.

Step 2 [Calibrating invariants]: In this step data is input to monitors. A monitor refers to a Python implementation of the invariant. There is one monitor for each invariant. Ideally all monitors must return "True" against D_N. However, as it is a physical system, that is not the case. Also, at least one monitor must return "False" when evaluated against A_N. Hence, at the end of this step, monitors are tuned to minimize false negatives while keeping false positives at a minimum.

Step 3 [Evaluating invariants]: A_N is input to the newly tuned monitors. The outcome of evaluating the monitors is

recorded and saved (set A). This set consists of the outcome of evaluating each monitor, i.e., true or false, on each data point over the data in A_N.

Step 4 [Parameter tuning, model calibration]: D_N is input to the model of WADI constructed as described in Section 5.1. The model is run in analysis mode. Again, all monitors must return "True" when evaluated against D_N and "False" when evaluated against A_N. The tuning parameters incorporated in the model are updated so that the false negatives are minimized while also keeping the false positive to a minimum.

Step 5 [Evaluating invariants in the model]: A_N is input to the model of WADI with the tuned parameters. The outcome of evaluating the monitors is recorded and saved (set B).

Step 6 [Model validation]: Ideally, the model is considered fully validated if the data in sets A and B are identical.

5.5. *Results and analysis*

Initially, in Step 2 of the validation procedure, the monitors triggered several false alarms. The false alarms were analyzed. It was found that the alarms were due to the following two reasons. One, an actuator was moving from one state to another. The state transition time was not accounted for while coding the corresponding monitor. Second, a small overshoot of the value from the desired range of the sensor led to a monitor evaluating to false. Manual analysis of the false alarms led to a modification of the monitors that led to false alarms.

False alarms due to state transitions were removed by adding delays to the monitors. Note that in Step 2, the monitors were coded by ignoring the state transition times in actuators. Then, at each instant when a state transition occurs, the monitors that consecutively evaluate to false are considered and a delay added; the delay is the maximum number of times the monitor led to consecutive false evaluations. For example, consider Invariant P1 listed above. According to this invariant, if level sensor 1_LT_001 reading is greater than 70, 1_MV_004 should be "Open" and 1_MV_001 "Closed". When 1_LT_001 indicates a level greater than 70, the control device interprets the data, and sends a signal to valves 1_MV_001 and 1_MV_004. Upon receiving the signal the valves change their state accordingly. This process takes a maximum of 12 seconds. Therefore, the delay for this invariant is set to 12 seconds. Once the calibration is completed, the model is run against the attack data to avoid false alarms.

Next, false alarms due to sensor reading overshoots were removed by adjusting the thresholds in the monitors. For example a threshold of 0.11 was added to the pH sensor in P1 (1_AIT_003_PV)

The output of Step 6 in the validation procedure indicates whether an attack is detected. This information is included in the rightmost column in Table 2. For example, attack 1 on process P1 is detected by invariant P1. In this attack, the attacker's intention is to overflow the raw water tank. In order to achieve this goal, the attacker attempts to change the state of actuator 1_MV_001 from "Close" to "Open" until the raw water tank overflows. However, invariant P1 shows that when the level of the raw water tank is greater than 70, both valves 1_MV_004 and 1_MV_001 must be closed. Thus, invariant P1 is violated and the attack detected.

Some attacks are not detected by the invariants. For example, attack 5 in stage 2 is not detected by any of the invariants. This is due to the lack of a complete set of invariants. The focus of the present study is the design of a methodology to create a model for a critical infrastructure. Hence, in this study we are not concerned with the creation of a complete set of invariants and the detection of all attacks; creation of invariants and their effectiveness in detecting attacks has been reported in the literature [2].

From Table 2 we note that a total of 12 out of 16 attacks were detected. A re-examination of these attacks revealed thet the attacks were not detected as the adversary ensures that the parameters do not exceed the normal operating values. Detecting such attacks requires more complex invariants that deal with time or pattern matching, and not only the current state of the plant. As this paper does not focus on attack detection, such invariants were not derived.

5.6. *Discussion*

Traditional approaches for modeling critical infrastructure and conducting attack detection studies are grounded in control theory such as in [31,37]. This paper presents an alternative, agent-based, approach to modeling critical infrastructure. The case study presented in this work does not include simulation which would require the use of control algorithms to be embedded in the model as a replacement for the three PLCs. While such simulation would be useful for validating control algorithms, and the model, the approach in this work makes use of data generated from a live plant. During plant operation the control algorithms resident in the PLCs are controlling the water distribution process in WADI. Hence, the impact of such algorithms is reflected in the data collected in Step 1 of the validation procedure. Thus, the procedure proposed here could be used as a first step in assessing the effectiveness of an attack detection mechanism once a model has been created.

One advantage of the agent-based approach over control theoretic approach appears to be in the ease with with the model can be created and the simplicity of the model. An agent based model need not include differential equations if the purpose of the model is to assess the effectiveness of an attack detection mechanism. Further, the relationship between the system components and the model is direct as seen from Table 1. This direct relationship is helpful in understanding the relationship among different components of the plant modeled. Further, the agent-based approach enables the integration of cyber components and the physical components in a plant. With such integration one can create a complete model of a CI. The communication between different PLCs can be considered as a commodity and the components would be similar to the actuators and sensors defined in this paper.

The proposed method was tested on an operational system. However, it is also possible to extent this model to other critical infrastructure like electrical systems. For example, the components of electrical systems can be classified as follows: wires as carrier; batteries and capacitors as resource; the current as the commodity; and resistors, switch and diode as actuator. Using this classification, an electrical system can be defined in a similar fashion as described in Section 4.

6. Summary and conclusions

In this paper a generic framework and an associated methodology are proposed. The methodology enables creation of a model of one or more critical infrastructure that may be interconnected. Such infrastructure includes water treatment and power generation plants. Components in a CI are categorized into 5-major classes namely commodity, carrier, resource, actuators and sensors. This categorization is not limited to a single type of infrastructure but works for any critical infrastructure. An operational real water distribution system was modeled using the proposed framework. To validate this model, experimental data from the system was used. First, the model was calibrated using the normal data obtained during plant operation. Then a set of attacks were launched on the system and the data generated used to detect the attacks. The model generated can interconnect multiple infrastructure even if they are of different types, for example, a water treatment plant and a power grid, thus pointing to the general applicability of the proposed approach. In the future we wish to work towards adding support for simulation of these systems using the models created.

Acknowledgment

This work was supported in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2015NCR-NCR003-001) and administered by the National Cybersecurity R&D Directorate.

Appendix

Table 3 – P1: sensors, actuators, and resources.

Sensors (m_s = P1)			
i_s	p_s	a_s	
1-LT-001	level	1-T-003	
1-FIT-001	flow	1-C-001	
1-AIT-001	conductivity	1-C-001	
1-AIT-002	turbidity	1-C-001	
1-AIT-003	pH	1-C-001	
1-AIT-004	ORP	1-C-001	
1-AIT-005	total residual chlorine	1-C-001	
Actuators (m_{ac} = P1)			
i_{ac}	p_{ac}	a_{ac}	
1-MV-001	status	1-C-001	
1-MV-002	status	1-C-013	
1-MV-003	status	1-C-012	
1-MV-004	status	1-C-003	
1-MV-005	status	1-C-002	
1-P-001	NaOCl	1-C-008	
1-P-002	NaOCl	1-C-009	
1-P-003	NH4Cl	1-C-010	
1-P-004	NH4Cl	1-C-011	
1-P-005	flow	1-C-005	
1-P-006	flow	1-C-006	
Resource (m_r = P1)			
i_r	S_r	I_r	O_r
1-T-001			1-C-008
1-T-001			1-C-008
1-T-003	1-LT-001	1-C-001	1-C-005
			1-C-013
1-T-004	1-LT-001	1-C-001	1-C-005
			1-C-012

Table 4 – P1: Carriers (m_c = P1).

i_c	S_c	AC_c	I_c	O_c
1-C-001	1-FIT-001	1-MV-001	SUTD Incoming	1-T-003
	1-AIT-001		1-C-004	1-T-004
	1-AIT-002		1-C-005	
	1-AIT-003		1-C-008	
	1-AIT-004		1-C-009	
	1-AIT-005		1-C-010	
			1-C-011	
1-C-002		1-MV=005	SWaT RO Permeate	1-C-001
			1-C-003	
1-C-003		1-MV-004	SWaT Raw Water Tank	1-C-002
1-C-004			Return Water	1-C-001
1-C-005			1-T-003	1-C-006
			1-T-004	1-C-007
1-C-006		1-P-005	1-C-005	Elevated Reservoir
1-C-007		1-P-006	1-C-005	Elevated Reservoir
1-C-008		1-P-001	1-T-001	1-C-001
1-C-009		1-P-002	1-C-008	1-C-001
1-C-010		1-P-003	1-T-002	1-C-001
1-C-011		1-P-004	1-C-010	1-C-001
1-C-012		1-MV-003	1-T-004	Ground
1-C-013		1-MV-002	1-T-003	Ground

Table 5 – P2a: sensors, actuators, and resources.

Sensors (m_s = P2a)			
i_s	p_s	a_s	
2-LT-001	level	2-T-001	
2-LT-001	level	2-T-002	
2-LS-001	level	2-T-003	
2-LS-002	level	2-T-004	
2A-AIT-001	conductivity	2A-C-006	
2A-AIT-002	turbidity	2A-C-006	
2A-AIT-003	pH	2A-C-006	
2A-AIT-004	ORP	2A-C-006	
Actuators (m_{ac} = P2a)			
i_{ac}	p_{ac}	a_{ac}	
2-MV-001	status	2A-C-002	
2-MV-002	status	2A-C-004	
2-MV-003	status	2A-C-003	
2-MV-004	status	2A-C-005	
2-MV-005	status	2A-C-007	
2-MV-006	status	2A-C-006	
2-P-001	status	2A-C-008	
2-P-002	status	2A-C-009	
Resource (m_r = P2a)			
i_r	S_r	I_r	O_r
2-T-001	2-LT-001	2A-C-002	2A-C-004
			2A-C-010
2-T-002	2-LT-002	2A-C-003	2A-C-005
			2A-C-011
2-T-003	2-LS-001		2A-C-008
2-T-004	2-LS-002		2A-C-009

Table 6 – P2a: carriers (m_c = P2a).

i_c	S_c	AC_c	I_c	O_c
2A-C-001	2-FIT-001		Raw Water Transfer- -Pump Skid	2A-C-002 2A-C-003
2A-C-002		2-MV-001	2A-C-001	2-T-001
2A-C-003		2-MV-003	2A-C-001	2-T-002
2A-C-004		2-MV-002	2-T-001	2A-C-006
2A-C-005		2-MV-004	2-T-002	2A-C-006
2A-C-006	2A-AIT-001 2A-AIT-002 2A-AIT-003 2A-AIT-004		2A-C-004 2A-C-005	Inline Booster Station
2A-C-007		2-MV-005	2A-C-006	Gravity Flow
2A-C-008		2-P-001	2-T-003	2A-C-006
2A-C-009		2-P-002	2-T-004	2A-C-006
2A-C-010			2-T-001	Ground
2A-C-011			2-T-002	Ground

Table 7 – P2b: sensors and actuators.

Sensors (m_s = P2b)		
i_s	p_s	a_s
2-FIT-002	flow	2B-C-001
2-FIT-003	flow	2B-C-009
2-PIT-003	pressure	2B-C-009
2B-AIT-001	conductivity	2B-C-001
2B-AIT-002	turbidity	2B-C-001
2B-AIT-003	pH	2B-C-001
2B-AIT-004	ORP	2B-C-001
Actuators (m_{ac} = P2b)		
i_{ac}	p_{ac}	a_{ac}
2-MV-008	status	2B-C-003
2-MV-009	status	2B-C-001
2-MCV-007	status	2B-C-002
2-MCV-008	status	2B-C-005
2-P-003	status	2B-C-007
2-P-004	status	2B-C-008

Table 8 – Carriers (m_c = P2b).

i_c	S_c	AC_c	I_c	O_c
2B-C-001	2-FIT-002 2B-AIT-001 2B-AIT-002 2B-AIT-003 2B-AIT-004		Gravity Flow	Gravity Feed Inlet 2B-C-002
2B-C-002		2-MCV-007	2B-C-001	2B-C-003
2B-C-003		2-MV-008	2B-C-002	Drain 2B-C-004
2B-C-004		2-MV-002	2B-C-003	Return
2B-C-005		2-MCV-008	2B-C-001	Ground
2B-C-006			Inlet Booster Station	2B-C-007 2B-C-008
2B-C-007		2-P-003	2B-C-006	2B-C-009
2B-C-008		2-P-004	2B-C-006	2B-C-009
2B-C-009	2-FIT-003		2B-C-007 2B-C-008	Booster Inlet

Table 9 – P2c: sensors, actuators, and resources.

Sensors (m_s = P2c)			
i_s	p_s	a_s	
2-LS-101	flow	2-T-101	
2-LS-201	flow	2-T-201	
2-LS-301	flow	2-T-301	
2-LS-401	flow	2-T-401	
2-LS-501	flow	2-T-501	
2-LS-601	flow	2-T-601	
Actuators (m_{ac} = P2c)			
i_{ac}	p_{ac}	a_{ac}	
2-MCV-101	status	2C-C-102	
2-FQ-101	flow	2C-C-102	
2-SV-101	status	2C-C-103	
2-MV-101	status	2C-C-104	
2-MCV-201	status	2C-C-202	
2-FQ-201	flow	2C-C-202	
2-SV-201	status	2C-C-203	
2-MV-201	status	2C-C-204	
2-MCV-301	status	2C-C-302	
2-FQ-301	flow	2C-C-302	
2-SV-301	status	2C-C-303	
2-MV-301	status	2C-C-304	
Actuators (m_{ac} = P2c)			
2-MCV-401	status	2C-C-402	
2-FQ-401	flow	2C-C-402	
2-SV-401	status	2C-C-403	
2-MV-401	status	2C-C-404	
2-MCV-501	status	2C-C-502	
2-FQ-501	flow	2C-C-502	
2-SV-501	status	2C-C-503	
2-MV-501	status	2C-C-504	
2-MCV-601	status	2C-C-602	
2-FQ-601	flow	2C-C-602	
2-SV-601	status	2C-C-603	
2-MV-601	status	2C-C-604	
Resource (m_r = P2c)			
i_r	S_r	I_r	O_r
2-T-101	2-LS-101	2C-C-102	2C-C-104
		2C-C-103	2C-C-105
2-T-201	2-LS-201	2C-C-202	2C-C-204
		2C-C-203	2C-C-205
2-T-301	2-LS-301	2C-C-302	2C-C-304
		2C-C-303	2C-C-305
2-T-401	2-LS-401	2C-C-402	2C-C-404
		2C-C-403	2C-C-405
2-T-501	2-LS-501	2C-C-502	2C-C-504
		2C-C-503	2C-C-505
2-T-601	2-LS-601	2C-C-602	2C-C-604
		2C-C-603	2C-C-605

Table 10 – P2c: carriers.

i_c	AC_c	I_c	O_c
2C-C-001		Gravity Feed Inlet	2C-C-101
		Booster Inlet	2C-C-201
			2C-C-301
			2C-C-401
			2C-C-501
			2C-C-601
2C-C-002		2C-C-104	Return Line
		2C-C-204	
		2C-C-304	
		2C-C-404	
		2C-C-504	
		2C-C-604	
2C-C-101		2C-C-001	2C-C-102
			2C-C-103
2C-C-102	2-MCV-101	2C-C-101	2-T-101
	2-FQ-101		
2C-C-103	2-SV-101	2C-C-101	2-T-101
2C-C-104	2-MV-101	2-T-101	2C-C-002
2C-C-105		2-T-101	Ground
2C-C-201		2C-C-001	2C-C-202
			2C-C-203
2C-C-202	2-MCV-201	2C-C-201	2-T-201
	2-FQ-201		
2C-C-203	2-SV-201	2C-C-201	2-T-201
2C-C-204	2-MV-201	2-T-201	2C-C-002
2C-C-205		2-T-201	Ground
2C-C-301		2C-C-001	2C-C-302
			2C-C-303
2C-C-302	2-MCV-301	2C-C-301	2-T-301
	2-FQ-301		
2C-C-303	2-SV-301	2C-C-301	2-T-301
2C-C-304	2-MV-301	2-T-301	2C-C-002
2C-C-305		2-T-301	Ground
2C-C-401		2C-C-001	2C-C-402
			2C-C-403
2C-C-402	2-MCV-401	2C-C-401	2-T-401
	2-FQ-401		
2C-C-403	2-SV-401	2C-C-401	2-T-401
2C-C-404	2-MV-401	2-T-401	2C-C-002
2C-C-405		2-T-401	Ground
2C-C-501		2C-C-001	2C-C-502
			2C-C-503
2C-C-502	2-MCV-501	2C-C-501	2-T-501
	2-FQ-501		
2C-C-503	2-SV-501	2C-C-501	2-T-501
2C-C-504	2-MV-501	2-T-501	2C-C-002
2C-C-505		2-T-501	Ground
2C-C-601		2C-C-001	2C-C-602
			2C-C-603
2C-C-602	2-MCV-601	2C-C-601	2-T-601
	2-FQ-601		
2C-C-603	2-SV-601	2C-C-601	2-T-601
2C-C-604	2-MV-601	2-T-601	2C-C-002
2C-C-605		2-T-601	Ground

Table 11 – Linking P2a, P2b, P2c.

Agent 1		Agent 2		Node
Agent	Model	Agent	Model	
Raw Water Transfer Pump Skid	P2	Raw Water Transfer Pump Skid	P2a	
Gravity Flow	P2a	Gravity Flow	P2b	Gravity Flow
Inline Booster Station	P2a	Inline Booster Station	P2b	Inline Booster Station
Gravity Feed Inlet	P2b	Gravity Feed Inlet	P2c	Gravity Feed Inlet
Booster Inlet	P2b	Booster Inlet	P2c	Booster Inlet
Drain	P2b	Drain	P2	
Return	P2b	Return	P2	
Return Line	P2b	Return Line	P2	

Table 12 – P3: sensors, actuators, and resources.

Sensors (m_s = P3)			
i_s	p_s	a_s	
3-LT-001	level	3-T-002	
3-FIT-001	flow	3-C-001	
3-AIT-001	conductivity	3-C-001	
3-AIT-002	turbidity	3-C-001	
3-AIT-003	pH	3-C-001	
3-AIT-004	ORP	3-C-001	
3-AIT-005	total residual chlorine	3-C-001	
3-LS-001	level	3-T-001	
Actuators (m_{ac} = P3)			
i_{ac}	p_{ac}	a_{ac}	
3-MV-001	status	3-C-007	
3-MV-002	status	3-C-005	
3-MV-003	status	3-C-006	
3-P-001	NaHSO3	3-C-008	
3-P-002	NaHSO3	3-C-009	
3-P-003	flow	3-C-003	
3-P-004	flow	3-C-004	
Resource (m_r = P3			
i_r	S_r	I_r	O_r
3-T-001	3-LS-001		3-C-009
3-T-002	3-LT-001	3-C-001	3-C-002
			3-C-007

Table 13 – P3: carriers (m_c = P3).

i_c	S_c	AC_c	I_c	O_c
3-C-001	3-FIT-001		Return Water	3-T-002
	3-AIT-001		3-C-008	
	3-AIT-002		3-C-009	
	3-AIT-003			
	3-AIT-004			
	3-AIT-005			
3-C-002			3-T-002	3-C-003
			3-C-003	3-C-004
3-C-003		3-P-003	3-C-002	3-C-005
				3-C-006
3-C-004		3-P-004	3-C-002	3-C-005
				3-C-006
3-C-005		3-MV-002	3-C-003	Return Water Tank
			3-C-004	
3-C-006		3-MV-003	3-C-003	Ground
			3-C-004	
3-C-007		3-MV-001	3-T-002	Ground
3-C-008		3-P-001	3-T-001	3-C-001
3-C-009		3-P-002	3-C-008	3-C-001

Table 14 – Linking P1, P2, P3.

Agent 1		Agent 2		Node
Agent	**Model**	**Agent**	**Model**	
SWaT RO Permeate	Wadi	SWat RO Permeate	P1	
SWaT Raw Water Tank	Wadi	SWaT Raw Water Tank	P1	
SUTD Incoming	Wadi	SUTD Incoming	P1	
Elevated Reservoir	P1	Raw Water Transfer Pump Skid	P2	ER - RWTPS
Return	P2	Return Line	P3	R - RL
Return Line	P2	Return Line	P3	RL - RL
Raw Water Tank	P3	Return Water	P1	RWT - RW
Ground	P1	Ground	Wadi	
Ground	P2	Ground	Wadi	
Drain	P1	Ground	Wadi	
Ground	P3	Ground	Wadi	

REFERENCES

[1] M. Abrams, J. Weiss, Malicious control system cyber security attack case study–Maroochy Water Services, Australia, Technical Report, The Mitre Corporation, McLean, VA, 2008.

[2] S. Adepu, A. Mathur, Distributed detection of single-stage multipoint cyber attacks in a water treatment plant, in:Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIA CCS '16, ACM, New York, NY, USA, 2016, pp. 449–460.

[3] S. Adepu, A. Mathur, Generalized attacker and attack models for Cyber-Physical Systems, in:Proceedings of the 40th Annual International Computers, Software & Applications Conference, Atlanta, USA, IEEE, Washington, D.C., USA, 2016, pp. 283–292.

[4] S. Adepu, A. Mathur, Introducing cyber security at the design stage of public infrastructures: A procedure and case study, in:Complex Systems Design & Management Asia, Springer, New York, USA, 2016, pp. 75–94.

[5] S. Adepu, A. Mathur, An investigation into the response of a water treatment system to cyber attacks, in:Proceedings of the 17th IEEE High Assurance Systems Engineering Symposium, Orlando, 2016, pp. 1–8.

[6] S. Adepu, A. Mathur, Using process invariants to detect cyber attacks on a water treatment system, in:Proceedings of the 31st International Conference on ICT Systems Security and Privacy Protection - IFIP SEC 2016 (IFIP AICT series), Springer, New York, USA, 2016, pp. 91–104.

[7] S. Adepu, A. Mathur, Assessing the effectiveness of attack detection at a hackfest on industrial control systems, IEEE Transactions on Sustainable Computing (2018).

[8] S. Adepu, A. Mathur, Distributed attack detection in a water treatment plant: Method and case study, IEEE Transactions on Dependable and Secure Computing (2018).

[9] S. Adepu, A. Mathur, J. Gunda, S. Djokic, An agent-based framework for simulating and analysing attacks on cyber physical systems, in:International Conference on Algorithms and Architectures for Parallel Processing, Springer, 2015, pp. 785–798.

[10] S. Adepu, G. Mishra, A. Mathur, Access control in water distribution networks: A case study, in:2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), IEEE, 2017, pp. 184–191.

[11] T. Adolphi, C. Nagel, T. Kolbe, Integrated 3d modeling of multi-utility networks and their interdependencies for critical infrastructure analysis, in:Advances in 3D Geo-Information Sciences, Springer, 1970, pp. 1–20.

[12] C. M. Ahmed, V. R. Palleti, A. P. Mathur, Wadi: A water distribution testbed for research in the design of secure cyber physical systems, in:Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER '17, ACM, New York, NY, USA, 2017, pp. 25–28.

[13] S. Amin, G. A. Schwartz, S. S. Sastry, Security of interdependent and identical networked control systems, Automatica 49(1) (2013) 186 – 192.

[14] D. Barton, E. D. Eidson, D. A. Schoenwald, K. Stamber, R. K. Reinert, Aspen-ee: An agent-based model of infrastructure interdependency, Sandia Report, SAND2000-2925 (2000).

[15] J. E. Bigger, M. G. Willingham, F. Krimgold, L. Mili, Consequences of critical infrastructure interdependencies: lessons from the 2004 hurricane season in florida, International Journal of Critical Infrastructures 5(3) (2009) 199–219.

[16] T. Brown, W. Beyeler, D. Barton, Assessing infrastructure interdependencies: the challenge of risk analysis for complex adaptive systems, International Journal of Critical Infrastructures 1(1) (2004) 108–117.

[17] V. Cardellini, E. Casalicchio, E. Galli, Agent-based modeling of interdependencies in critical infrastructures through uml, in:Proceedings of the 2007 Spring Simulation Multiconference - Volume 2, SpringSim '07, Society for Computer Simulation International, 2007, pp. 119–126.

[18] V. Cardellini, E. Casalicchio, S. Tucci, Agent-based modeling of web systems in critical information infrastructure (2006).

[19] S. N. Dorogovtsev, A. V. Goltsev, J. F. F. Mendes, Critical phenomena in complex networks, Reviews of Modern Physics 80(4) (2008) 1275–1335.

[20] D. D. Dudenhoeffer, M. R. Permann, M. Manic, Cims: A framework for infrastructure interdependency modeling and analysis, in:Proceedings of the 2006 Winter Simulation Conference, 2006, pp. 478–485.

[21] S. Hasan, G. Foliente, Modeling infrastructure system interdependencies and socioeconomic impacts of failure in extreme events: emerging r&d challenges, Natural Hazards 78(3) (2015) 2143–2168.

[22] C. Heracleous, P. Kolios, C. G. Panayiotou, G. Ellinas, M. M. Polycarpou, Hybrid systems modeling for critical infrastructures interdependency analysis, Reliability Engineering & System Safety 165 (2017) 89 – 101.

[23] E. E. L. II, J. E. Mitchell, W. A. Wallace, Restoration of services in interdependent infrastructure systems: A network flows approach, IEEE Transactions on Systems, Man, and

Cybernetics, Part C (Applications and Reviews) 37(6) (2007) 1303–1317.
[24] Y. Kajitani, S. Sagai, Modelling the interdependencies of critical infrastructures during natural disasters: a case of supply, communication and transportation infrastructures, International Journal of Critical Infrastructures 5(1–2) (2009) 38–50.
[25] E. Kang, S. Adepu, D. Jackson, A. P. Mathur, Model-based security analysis of a water treatment system, in:In Proceedings of 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'16), 2016, pp. 1–7.
[26] N. Kollikkathara, H. Feng, D. Yu, A system dynamic modeling approach for evaluating municipal solid waste generation, landfill capacity and related cost management issues, Waste Management 30(11) (2010) 2194 – 2203. Special Thematic Section: Sanitary Landfilling.
[27] D. F. Laefer, A. Koss, A. Pradhan, The need for baseline data characteristics for gis-based disaster management systems, Journal of Urban Planning and Development 132(3) (2006) 115–119.
[28] J.-C. Laprie, K. Kanoun, M. Kaâniche, Modelling interdependencies between the electricity and information infrastructures, in:Computer Safety, Reliability, and Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 54–67.
[29] Q. Lin, S. Adepu, S. Verwer, A. Mathur, Tabor: A graphical model-based approach for anomaly detection in industrial control systems, in:Proceedings of the 2018 on Asia Conference on Computer and Communications Security, ACM, 2018, pp. 525–536.
[30] R. Lipovsky, New wave of cyber attacks against Ukrainian power industry, 2016, http://www.welivesecurity.com/2016/01/11.
[31] Y. Liu, P. Ning, M. K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Transactions on Information and System Security (TISSEC) 14(1) (2011) 13.
[32] G. Oliva, S. Panzieri, R. Setola, Agent-based inputoutput interdependency model, International Journal of Critical Infrastructure Protection 3(2) (2010) 76 – 82.
[33] T. D. O'rourke, A. J. Lembo, L. K. Nozick, Lessons learned from the world trade center disaster about critical utility systems, in:Beyond September 11th: An Account of Post-Disaster Research, Natural Hazards Research and Applications Information Center, Boulder, 2003, pp. 269–290.
[34] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Reliability Engineering & System Safety 121(Supplement C) (2014) 43 – 60.
[35] M. Ouyang, Critical location identification and vulnerability analysis of interdependent infrastructure systems under spatially localized attacks, Reliability Engineering & System Safety 154 (2016) 106 – 116.
[36] V. R. Palleti, J. V. Joseph, A. Silva, A contribution of axiomatic design principles to the analysis and impact of attacks on critical infrastructures, International Journal of Critical Infrastructure Protection (2018). https://doi.org/10.1016/j.ijcip.2018.08.007.
[37] F. Pasqualetti, F. Dorfler, F. Bullo, Attack detection and identification in cyber-physical systems, IEEE Transactions on Automatic Control 58(11) (2013) 2715–2729.
[38] G. E. A. Patterson, Sean A., Identification of critical locations across multiple infrastructures for terrorist actions, Reliability Engineering and System Safety 92 (2007) 1183–203.
[39] H. M. Paynter, Analysis and design of engineering systems, MIT Press, Cambridge, MA, USA, 1961.
[40] E. G. Quijano, D. R. Insua, J. Cano, Critical networked infrastructure protection from adversaries, Reliability Engineering & System Safety 152 (2016) 137 – 150.
[41] H. Sandberg, S. Amin, K. H. Johansson, Cyberphysical security in networked control systems: An introduction to the issue, IEEE Control Systems 35(1) (2015) 20–23.
[42] G. Satumtira, L. Duenas-Osorio, Synthesis of modeling and simulation methods on critical infrastructure interdependencies research, in:K. Gopalakrishnan, S. Peeta (Eds.), Sustainable and Resilient Critical Infrastructure Systems: Simulation, Modeling, and Intelligent Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 1–51.
[43] D. M. Simpson, C. B. Lasley, T. D. Rockaway, T. Weigel, Understanding critical infrastructure failure: examining the experience of biloxi and gulfport, mississippi after hurricane katrina, International Journal of Critical Infrastructures 6(3) (2010) 246–276.
[44] R. Stapelberg, Infrastructure systems interdependencies and risk informed decision making (ridm): Impact scenario analysis of infrastructure risks induced by natural, technological and intentional hazards, Journal of Systemics, Cybernetics and Informatics 6(5) (2008) 21–7.
[45] S. Sultana, Z. Chen, Modeling flood induced interdependencies among hydroelectricity generating infrastructures, Journal of Environmental Management 90(11) (2009) 3272 – 3282.
[46] S. TANAKA, Accident at the fukushima dai-ichi nuclear power stations of tepco–outline and lessons learned, Proceedings of the Japan Academy, Series B 88(9) (2012) 471–484.
[47] J. R. Thompson, D. Frezza, B. Necioglu, M. L. Cohen, K. Hoffman, K. Rosfjord, Agent-based modelling of interdependent critical infrastructures, International Journal of System of Systems Engineering 2 (2010) 60–75.
[48] J. R. Thompson, D. Frezza, B. Necioglu, M. L. Cohen, K. Hoffman, K. Rosfjord, Interdependent critical infrastructure model (icim): An agent-based model of power and water infrastructure, International Journal of Critical Infrastructure Protection 24 (2019) 144 – 165.
[49] W. A. Wallace, D. Mendonca, E. E. Lee, J. E. Mitchell, J. H. Chow, Managing disruptions to critical interdependent infrastructures in the context of the 2001 world trade center attack, in:Beyond September 11: An account of post-disaster research, 2002, pp. 165–198.
[50] S. Weinberger, Computer security: Is this the start of cyberwarfare?, Nature 174 (2011) 142–145.
[51] E. Zio, Challenges in the vulnerability and risk analysis of critical infrastructures, Reliability Engineering & System Safety 152 (2016) 137 – 150.