

Faults and Cyber Attacks Detection in Critical Infrastructures

Yannis Soupionis, Stavros Ntalampiras and Georgios Giannopoulos

European Commission, Joint Research Centre (JRC)
Institute for the Protection and Security of the Citizen (IPSC)
Security Technology Assessment Unit
Via E. Fermi 2749, 21027, Ispra (VA), Italy
{yannis.soupionis, stavros.ntalampiras, georgios.giannopoulos}@jrc.ec.europa.eu

Abstract. The explosive growth of the Internet has introduced advanced services facilitating our every day life. On the other hand it created increased dependence on computer systems which may be seen as an additional source of vulnerability to disruption scenarios due to both physical and cyber-based incidents. In this paper we study the case of Critical Infrastructures (CIs), and especially power grid systems, which is one of the variety of services that nowadays relies on computers and the Internet for its operation. We design an experimental platform consisting of a power grid simulator and a cyber network emulator. This paper proposes a combinatorial method for automatic detection and classification of faults and cyber-attacks occurring on the power grid system when there is limited data from the power grid nodes due to cyber implications. The efficiency of the proposed method is demonstrated via an extensive experimental phase measuring the false positive rate, false negative rate and the delay of the detections.

Keywords: Critical infrastructures; cyber security; DDoS; fault diagnosis; linear time-invariant modeling;

1 Introduction

Modern Critical Infrastructures (CI), e.g. power plants, water distribution networks and transport systems, rely on Information and Communication Technologies (ICT) for their operation since ICT can lead to cost reduction as well as greater efficiency, flexibility and interoperability between components. In the past CIs were isolated environments and used proprietary hardware and protocols, a logic motivated by the need to limit or even eliminate the impact of any type of potential threats. However the specific logic comes with many functional limitations, e.g. higher installation, maintenance and operational cost coming from not infrastructure sharing.

Nowadays, CIs and more specifically Distributed Control Systems (DCS) are exposed to significant cyber-threats, a fact that has been highlighted by many

studies on the security of Supervisory Control And Data Acquisition (SCADA) systems [1], [2], [3]. For example, the Stuxnet worm [4] is the first malware that is specifically designed to attack industrial control systems.

In this paper we explain the development of a method able to automatically detect and classify faults and/or cyber-attacks based solely on the provided communication data between the SCADA systems and the local Programmable Logic Controllers (PLCs) of one of the most important classes of CIs, *power grids*. Furthermore we are able to decide on the kind of the cyber-physical implication even when there is limited data/information provided. This article comprises a preliminary effort towards identifying failures in interdependent CIs [5], [6] and obtaining situational awareness.

The paper is organized as follows: Section 2 describes the related work while in Section 3 we present the main elements of a distributed control system, the communication links between its components and our implementation framework. In section 4 we explain the proposed detection and classification method and in section 5 we describe the examined cyber-physical implications. Section 6 illustrates the way our method can be applied onto a power grid and describes the experimental results. Finally Section 7 concludes this work and identifies some open subjects for future work.

2 Related work

In this section, we briefly survey some techniques and approaches, which focus on either cyber attack or fault detection studies on cyber-physical infrastructures.

2.1 Cyber attacks

Faults and/or attacks occurring on the cyber layer comprise the main implication of an interdependent cyber-physical system. It is [7] clearly illustrated that the existing techniques are not adequate to address the series of new security challenges and threats posed by highly complex environments such as the smart grid. In [8], the importance of this issue is stressed out and some high level guidelines are proposed in order to prevent, mitigate, and tolerate cyber attacks. Even though, they present the significance of the cyber infrastructure security, they do not propose a detection or mitigation process.

Finally, a quite interesting research field is the one focusing on false injection attacks, a class of the integrity attacks. Various techniques have been developed to detect and identify erroneous measurements [9, 10]. Even though those approaches are effective, they do not take into account the interdependent nature of cyber-physical infrastructures. Moreover, they implement those attacks on a theoretical level and evaluate these methods by simulating all the aspects of the elements of the smart grid network. In our case we manage to implement not only the cyber attacks but also the physical faults in a emulated environment providing a quite representative depiction of real infrastructures.

2.2 Fault detection

Fault diagnosis systems usually operate on data coming from sensor networks and they can be divided into approaches exploiting information coming either from a *single* sensor or a *set* of sensors (a detailed review of the fault-detection literature can be found in [11] and [12]). The first kind of approaches relies on data coming from a single sensor including mainly limit checking [13], the basic principle of which is to detect a fault when the physical quantity under monitoring overcomes/drops below a predefined threshold.

Approaches using data coming from multiple sensors detect faults by exploiting potential redundancies and/or correlations existing within the data. Here we find two lines of thought: a) the first one exploits physical redundancy, i.e. redundant sensors [14], and b) the second one takes advantage of analytical redundancy based on the functional relationships existing among different, but correlated, quantities [15].

In the present work we aim to provide elements towards addressing this gap for the specific case of ICT/Energy interdependent systems.

3 Distributed Control Systems

In this section we provide a brief description of the DCS architecture used in this work. We present the standardized architecture and subsequently our implementation of the simulation/emulation framework.

The ICT infrastructure of a power grid system is consisted mainly by the automation control. It consists of control centers, which supervise the operation of the substations. The physical layer is composed of field devices, like sensors, meters, phase measuring units, which send raw information to the first layer. There we have mainly the PLCs and lower-level distributed controls. The higher layers contains more advanced controlling processes as SCADA server. Moreover, in order to deliver electrical power from producers to consumers in a cost-effective way, the central power grid operators have to exchange information with various organization and devices, such as Independent System Operators and Energy market.

The used implementation framework a) simulates the physical components of a power grid and b) emulates the cyber elements. The models of the physical systems are developed in Matlab Simulink, from which we generate the corresponding 'C' code using Matlab Real Time Workshop. The generated code is then executed in real time and interacts with the real components of our emulation testbed. From a technical point of view, real-time simulation of IEEE grid models is implemented in the Assessment platform for Multiple Interdependent Critical Infrastructures (AMICI) [16], which is based on Matlab open-source libraries, i.e. MatPower [17] and MatDyn [18]. The power grid employed in this experiment is the well-known IEEE 9-bus model (see Fig. 1 for its graphical representation). It includes a) 3 bus generators, b) 3 substations that deliver power to connected loads through transmission lines, and c) 3 dedicated buses playing

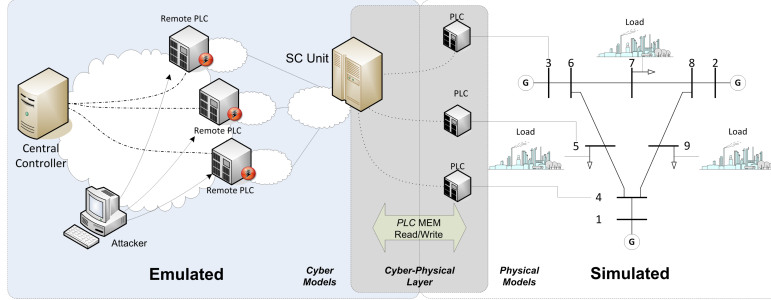


Fig. 1. The overall architecture of the simulation & emulation framework used in this work.

the consumer role. Finally, the sampling rate of the simulated model is 20ms which is the time to calculate its parameters including the interaction with the emulated environment.

The testbed of the DCS cyber part (SCADA servers, corporate network, etc.) facilitating the emulation of the ICT infrastructure is based on Emulab [19], which is a well-established network emulator. The software for using the Emulab architecture was developed in our laboratory [20]. To this end, we can map and replicate various network physical infrastructures, which contain physical elements as routers and switches including the links between them. These elements are critical for acquiring accurate measurements and developing testbeds which match quite well the real-world conditions. The communication between the remote PLCs and the controlling units is based on the Modbus over TCP protocol. However, another protocol can be easily integrated due to the modularity provided by our implementation.

4 The Fault Detection Method

This section explains the fault detection method designed for the case of critical cyber-physical networks. It is comprised of two different algorithms running concurrently and combined with an AND logic meaning that the system detects a fault only when both methods detect one for a specific datum (see Fig. 2). The AND logic was chosen due to the severe need of CIs to avoid false alarms. The motivation behind their fusion is the fact that the methods are heterogeneous in the sense that the first one operates directly on the domain of the CI variable while the second measures the discrepancy between the actual data value and the one predicted by an Linear-Time Invariant (LTI) model.

Let us consider an energy monitoring framework comprised of N buses and K generators each of which provides a time-series datastream. Denote by $X_i : \mathbb{N} \rightarrow \mathbb{R}$ the stream of data acquired by the i -th bus and $X_j : \mathbb{K} \rightarrow \mathbb{R}$ the stream of data acquired by the j -th generator.

Let $O_{i,T_0} = \{X_i(t), t = 1, \dots, T_0\}$ and $O_{j,T_0} = \{X_j(t), t = 1, \dots, T_0\}$ be the data sequence of the i -th bus and the j -th generator respectively. Finally,

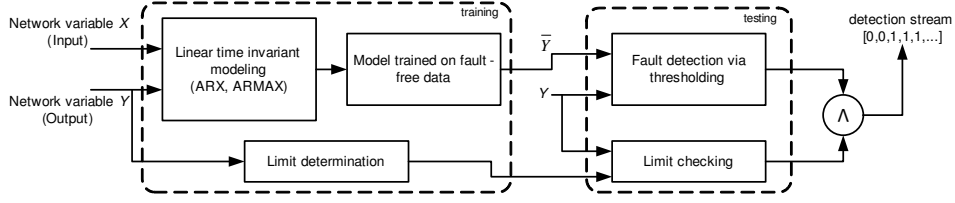


Fig. 2. The block diagram of the proposed fault detection method for Critical Infrastructures. The algorithm detects potentially faulty data coming from network variable Y (1 stands for detection and 0 for nominal data). In our case Y corresponds to the V_i output per bus and X to the real power demand.

let us assume that at an unknown time instant $T^* > T_0$ a fault occurs in the datastreams while no assumption is made about its magnitude or time profile. Specifically for the IEEE network model X_i corresponds to the V_i output per bus and X_j to the real power demand.

We designed two methods trying to detect a fault appearing the data sequence:

- *Limit checking*: This method checks whether the datastream of interest is within a bandwidth specified by a maximum and a minimum value: $[U_{bound}, L_{bound}]$. In case an incoming datum $O_t, t > T_0$ is out of the bandwidth determined during the training phase, it is marked as faulty. This process is explained in Algorithm 1.

1. Identify the maximum and lower values of the training sequence

$$O_{i,T_0} = \{X_j(t), t = 1, \dots, T_0\} \text{ as}$$

$$U_{bound} = \max(O_{i,1\dots T_0}), L_{bound} = \min(O_{i,1\dots T_0}) ;$$

repeat

2. $t=1$;

3. **if** $O_{(j,t)} > U_{bound} \vee O_{(j,t)} < L_{bound}$ **then**
 | $O_{(j,t)}$ contains data associated with a fault;

else

| $O_{(j,t)}$ contains data coming from the normally operating network;

end

4. $t = t + 1$;

until (1);

Algorithm 1: The limit checking fault detection algorithm.

- *LTI modeling*: This method models the relationships between the datastreams belonging to physical variables of the CI under study. The underlying assumption here is that the pattern of the relationship remains consistent when the system operates in a certain state (nominal, faulty, etc.). The proposed fault detection technique assumes that the relationship between two generic correlated datastreams i and j , j used to infer i , can be

described through an input-output dynamic model of the form

$$\begin{aligned} X_i(k) = f_\theta(X_i(k-1), X_i(k-2), \dots, X_i(k-k_i), \\ X_j(k), X_j(k-1), \dots, X_j(k-k_j)) \end{aligned}$$

where f is a linear function of autoregressive model with exogenous input (ARX) type in its parameters θ and k_i and k_j are the orders of the model. The model with the lowest reconstruction error is chosen.

The specific fault detection method relies on the discrepancy observed between the data simulated via the model trained on fault-free data and the actual one. When the discrepancy is over a threshold, the datum is marked as faulty. This process is given in Algorithm .2.

1. Find the model \mathbb{M} explaining the relationship between $X_i(k)$ and $X_j(k)$ with the lowest reconstruction error ;
 2. Apply \mathbb{M} on $O_{i,T_0} = \{X_i(t), t = 1, \dots, T_0\}$ and compute the estimated data values $\bar{O}_{j,T_0} = \{\bar{X}_i(t), t = 1, \dots, T_0\}$;
 3. $T_h = \max(|\bar{O}_{j,T_0} - O_{j,T_0}|)$;
- repeat**
2. $t=1$;
 3. **if** $|\bar{O}_{j,T_0} - O_{j,T_0}| > T_h$ **then**
 - | $O_{(j,t)}$ contains data associated with a fault;
 - else**
 - | $O_{(j,t)}$ contains data coming from the normally operating network;
 - end**
 4. $t = t + 1$;
- until** (1);

Algorithm 2: The model-based fault detection algorithm.

The overall method combines the methods demonstrated in Algorithms 1 and 2 via the *AND* logic as demonstrated in Fig. 2. The aim of the proposed fault detection method is to detect the occurred fault with the smallest latency and false positive and negative rates.

With respect to isolation and classification of a detection into fault, integrity attack or DDoS we rely on a distance matrix D including all the discrepancies which are observed between the actual data and the values predicted from the ARX model. More precisely, for each state we produce a distance matrix D :

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix},$$

where n is the number of buses included in the network and $d_{ij} = |\bar{O}_{j,T_0} - O_{j,T_0}|$ given a fault on BUS i . Each line is associated with a fault occurring on a BUS,

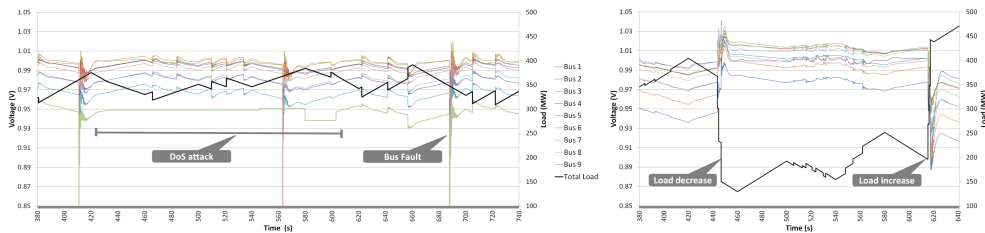


Fig. 3. Training session data graph (DoS attack on bus 9 & Bus 8 fault & load increase-decrease $\simeq 250MW$)

e.g. element d_{21} includes the discrepancy observed on the Voltage of BUS 1 when a fault on BUS 2 has occurred. The buses presenting faults comprise the column elements. During testing, the system gathers the observed distances and computes their difference to the ones computed during training (signatures). Finally the fault/DDoS/Integrity is identified based on the minimum distance criterion.

5 Cyber-Physical infrastructure implications

In this section we present the power grid implications we take into consideration:

1. Distributed Denial of Service Attacks: Unfortunately modern CIs have experienced a variety of DDoS attacks as described in [21]. For example, a DoS attack may flood the SCADA master or the RTU with valid protocol messages aiming at the saturation of the CPU computational power, memory and/or communication bandwidth. This situation could result in delay or inhibition of real-time data exchange. As a consequence, control center operators may fail to have a complete view of the electrical power grid system status which may lead to make incorrect decisions. During a DDoS attack only sparse data reach the controller, which keeps the last received value till a new one arrives. More specifically, $V_i(t) = V_i(t_0), t_0 < t < t_1$, where t_0 is the time that the last datum was received and t_1 is the time that the new datum may arrive (Fig. 3).

2. Integrity & replay attacks: These can be implemented either by affecting the power grid component/equipments which are responsible for distribution systems or by manipulating the exchanging protocol messages. The first situation concerns devices, which can be compromised. The latter affects known protocols, such as Modbus and DNP3 where the attacker can manipulate the protocol messages and send malicious data to the field device or the control center operator.

3. Fault: We take into account sudden losses of connectivity affecting one bus for a short period of time. This fault is supported by MatPower software and it is a common error in electrical nodes. The simulated fault is essentially a two-phase bus fault. It happens by changing the shunt susceptance of a bus and the fault can be cleared by resetting the susceptance to its original value.

6 Experimental setup and results

In this section we evaluate the proposed method for detection and classification of cyber and physical events in a DCS. Our evaluation procedures follows three distinct steps:

1. First we create a detailed case study of the virtual network topology and physical simulated power grid as described in section 3. We should state that the simulation step is 20ms.
2. The experiments are then initiated in order to collect data and train the detector. They are detailed in Section 6.1.
3. Finally, experimental scenarios are executed in order to evaluate our system's performance. It should be stated that the scenarios include situations where a node is under attack and at the same time a fault occurs.

Table 1. Training scenarios

Scenarios	Values (V in p.u.)	Description
<i>Normal conditions</i>	$0.9 \leq V_{busi} \leq 1.1$	The power grid operates smoothly
<i>Sudden load increase and decrease</i>	$Load_{alteration} > 250MW$	The power grid reaches marginal state
<i>Bus fault to i node</i>	$V_{busi} \simeq 0$	Bus is down for 0.2s
<i>DDoS attack to i node</i>	$V_{busi} \simeq V_{busi_{Lastreceived}}$	Bus data partially reaches central control
<i>DDoS attack & bus fault to i node</i>	$V_{busi} \simeq V_{busi_{Lastreceived}}$	Fault during a DDoS attack

6.1 Detection training scenarios

The detection method presented in section 4 needs a training process before being able to provide real-time results. During the training process a set of scenarios on the aforementioned experimental environment were simulated. The produced dataset was provided to the detection mechanism for creating a schema of the possible experimental environment states.

The experimental scenarios executed for the training are (Table 1) :

1. scenario setting the normal state of the power grid. Here, limited fluctuations in consumed load are present without faults and/or attacks.
2. scenario where either excessive additional consumed load is needed or a sudden consumed load drop occurs. In these situations the smart grid may reach or even surpass its boundary state regardless the absence of kind of implications (Fig. 3).

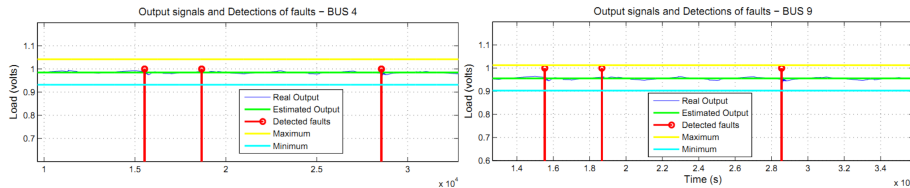


Fig. 4. A paradigm of the way the fault diagnosis framework operates in case of a DDoS attack on Bus 1. Bus 1 stops providing data and afterwards a fault occurs to it. We detect a fault, and by consulting the distance matrix D on neighbor Buses 4 and 9, we conclude that the fault occurred on Bus 1.

3. scenario about physical faults, as they were described in section 5. We injected faults to every bus of the smart grid in order to identify not only the specific bus's behavior but also the cascading effects on the other buses. This is a crucial information in order to detect faults when the controller is provided with limited information. An example of the smart grid fault state is depicted in Fig. 3.
4. scenario where a DDoS against a node's PLC for a limited period of time occurs (Section 5).
5. the last scenario included a two-step smart grid implication. A node is under a DDoS attack and during this period a fault occurs to a specific node. This scenario is very valuable because we train our system to identify faults not only by the data transmitted from the "faulty" node but also by analyzing the data from the neighbor nodes. An example of this state is depicted in Fig. 3.

6.2 Detection results

Here we describe the evaluation of the detection and identification capabilities of the proposed method. Extensive experiments were designed and conducted in order to assess the performance of the system in a reliable manner. To this end, three figures of merits have been defined, namely False Positive rate (FP), False Negative rate and Detection Delay (DD): it measures the time delay in detecting a change.

The detection framework is trained on data coming from the normal modality of total length of 10000 samples. The limit checking method uses the entire length for bound estimation $[U_{bound}, L_{bound}]$. The model-based method uses the scenarios in section 6.1 for training. The input is the exchanging data and the current state of the smart grid. This means that we have as an input the starting and finishing point of each aforementioned state. The needed samples for the model training for each state are less 8000, meaning less than 2.5 minutes of sampling data. The model order is determined by minimizing a robustified quadratic prediction error criterion which serves the computation of the ARX

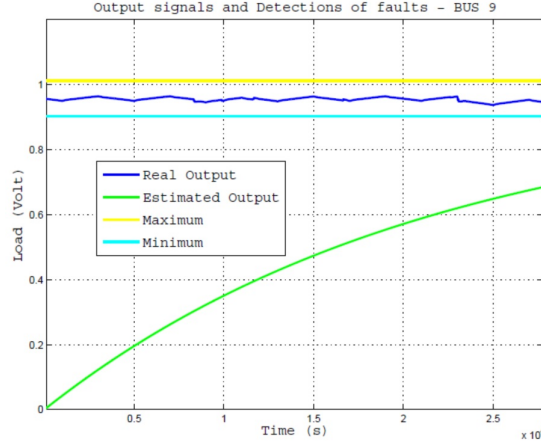


Fig. 5. A paradigm of the way the fault diagnosis framework operates in case of an integrity attack on Bus 9. The method detects the attack since the data coming from Bus 9 is not consistent with the data provided by the rest of the buses.

parameters. Finally, the model which provided the best performances was the following ARX(2,2):

$$X_i(t) = a_1 X_i(t-1) + a_2 X_i(t-2) + b_1 X_j(t-1) + b_2 X_j(t-2) \quad (1)$$

where $a_1 = 0.5$, $a_2 = 0.2$, $b_1 = 0.1$, $b_2 = 0.3$. The figures of merit are computed both on data coming from the nominal and faulty states. They are tabulated in Table 2.

The first experiment was a two events scenario: we initiated a DDoS against a node and in a few seconds we injected a fault to the same node. The main issue is that since there is extremely limited information from the faulty node, we have to find a way to identify that the fault was initiated from the specific node. In Fig. 4 we illustrate the method to identify the fault to one node by analyzing the input from its neighbor ones.

The second scenario was an integrity attack against a single node. The integrity attack compromised the node's PLC and force the sending of false data to the central control. The sent data was not corrupted, which would made their identification trivial, but it was based on true data acquired from another state of the machine. For example, the attacker has previously implemented a Man-in-the-Middle attack, copied the transmitted data and now is able to replicate it. In Fig. 5 we demonstrate the operation of the proposed fault diagnosis system when an integrity attack takes place.

We conducted experiments with most of the nodes and we claim that our method is able to predict the malicious state close to 98% (Table 2). In addition Table 2 provides the FPs, FNs and delays with respect to every faulty/attack situation. We observe that they are kept under quite low values while integrity

Table 2. The detection results of the proposed method. The figures of merit are averaged over the entire dataset.

Test data type	FP (%)	FN (%)	Detection Delay (# of samples)
<i>Nominal</i>	0	-	-
<i>Overload (fault-free)</i>	0.2	-	-
<i>Underload (fault-free)</i>	0.5	-	-
<i>Fault</i>	6.1	2.7	12
<i>DDoS</i>	7.1	1	4.2
<i>Integrity</i>	10.1	2.3	5.75

attacks are the ones detected with the largest delay. We infer that the performance proposed system is encouraging and effective given the complexity of the problem including two diverse and interdependent critical infrastructures.

7 Conclusions and further research

Power grids operation should be smooth and uninterrupted while data exchange with other infrastructures including central controller and energy markets is a fundamental prerequisite. In this paper, we provided an automatic method for detecting and classifying cyber implications and physical faults affecting a power grid infrastructure for enhancing the overall resilience. The method is able to identify the errors even when operating under adverse conditions. We are able to identify the power grid node state even when the node is unreachable or off-line or exchanging data is malformed. The method was evaluated not only on theoretical level but also in practice, by implementing a cyber physical infrastructure, which combines a simulated power grid and an emulated ICT network.

Further research includes a) cooperation with a power grid operator in order to evaluate the method on real-world datasets, b) design a more sophisticated fault detection algorithm able to 'understand' the state of larger networks using only a small part of the available incoming information, c) new intelligent cyber attacks (e.g. integrity), and d) analysis of the robustness of the proposed method when statistical variations appear in the datastreams.

References

1. C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for scada systems," *Power Systems, IEEE Transactions on*, vol. 23, no. 4, pp. 1836–1846, Nov 2008.
2. C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *Sys., Man and Cybern., Part A: Systems and Humans, IEEE Trans. on*, vol. 40, no. 4, pp. 853–865, July 2010.
3. C. Alcaraz, J. Lopez, "Wide-Area Situational Awareness for Critical Infrastructure Protection," *Computer*, vol.46, no.4, pp.30-37, 2013.

4. R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, May 2011.
5. E. Zio and G. Sansavini, "Modeling interdependent network systems for identifying cascade-safe operating margins," *Reliability, IEEE Trans.*, vol.60, no.1, pp. 94-101,2011.
6. Kotzanikolaou P., Theoharidou M., Gritzalis D., "Accessing n-order dependencies between critical infrastructures", *International Journal of Critical Infrastructures*, vol.9, no.1-2, pp.93-110, 2013.
7. Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber - physical security of a smart grid infrastructure," *Proc. of the IEEE*, vol. 100, no. 1, pp. 195–209, Jan 2012.
8. S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber - physical system security for the electric power grid," *Proc. of the IEEE*, vol. 100, no. 1, pp. 210–224, Jan 2012.
9. Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 13:1–13:33, Jun. 2011.
10. O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *Smart Grid, IEEE Transactions on*, vol. 2, no. 4, pp. 645–658, 2011.
11. R. Isermann, *Fault-diagnosis systems : an introduction from fault detection to fault tolerance*. Berlin: Springer, 2006.
12. V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, *Computers and Chemical Engineering*, no. 3, pp. 293–311.
13. W. A. Shewart, *Economic control of Quality of Manufactured Product*. New York: Van Nostrand Reinhold Co., 1931.
14. K. Goebel and W. Yan, "Correcting sensor drift and intermittency faults with data fusion and automated learning", *IEEE Systems*, vol.2, no.2, pp.189–197, 2008.
15. M.R. Napolitano, D.A. Windon, J.L. Casanova, M. Innocenti, and G. Silvestri, "Kalman Filters and Neural Network Schemes for Sensor Validation in Flight Control Systems," *Control Systems Technology, IEEE Trans.*, vol.6, no.5, pp. 596–611, 1998.
16. B. Genge, C. Siaterlis, and M. Hohenadel, "Amici: An assessment platform for multi-domain security experimentation on critical infrastructures," in *Critical Information Infrastructures Security*, 2013, pp. 228–239.
17. R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12–19, Feb 2011.
18. S. Cole and R. Belmans, "Matdyn, a new matlab-based toolbox for power system dynamic simulation," *Power Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 1129–1136, Aug 2011.
19. B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *OSDI02*. Boston, Dec 2002, pp. 255–270.
20. C. Siaterlis, A. Garcia, and B. Genge, "On the use of Emulab testbeds for scientifically rigorous experiments," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–14, 2012.
21. T. Bass, "Intrusion detection systems and multisensor data fusion," *Commun. ACM*, vol. 43, no. 4, pp. 99–105, Apr. 2000.