Reliability Engineering and System Safety 138 (2015) 73–81

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

A fault diagnosis system for interdependent critical infrastructures based on HMMs

Stavros Ntalampiras*, Yannis Soupionis, Georgios Giannopoulos

European Commission, Joint Research Center, Institute for the Protection and Security of the Citizen, Via E. Fermi, 2749, 21027 Ispra (VA), Italy

ARTICLE INFO

Article history:
Received 30 July 2014
Received in revised form
20 January 2015
Accepted 24 January 2015
Available online 2 February 2015

Keywords:
Critical infrastructure protection
Linear time invariant modeling
Hidden Markov model
Fault diagnosis
Cyber security
Cyber-attacks

ABSTRACT

Modern society depends on the smooth functioning of critical infrastructures which provide services of fundamental importance, e.g. telecommunications and water supply. These infrastructures may suffer from faults/malfunctions coming e.g. from aging effects or they may even comprise targets of terrorist attacks. Prompt detection and accommodation of these situations is of paramount significance.

This paper proposes a probabilistic modeling scheme for analyzing malicious events appearing in interdependent critical infrastructures. The proposed scheme is based on modeling the relationship between datastreams coming from two network nodes by means of a hidden Markov model (HMM) trained on the parameters of linear time-invariant dynamic systems which estimate the relationships existing among the specific nodes over consecutive time windows. Our study includes an energy network (IEEE 30 model bus) operated via a telecommunications infrastructure.

The relationships among the elements of the network of infrastructures are represented by an HMM and the novel data is categorized according to its distance (computed in the probabilistic space) from the training ones. We considered two types of cyber-attacks (denial of service and integrity/replay) and report encouraging results in terms of false positive rate, false negative rate and detection delay.

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

1. Introduction

Modern critical infrastructures (CI) include numerous elements for facilitating different functions of a society and its economy. A CI is an infrastructure, the smooth operation of which is essential to maintain the quality of life and safety of the citizens as well as it economic security. CIs include but are not limited to: telecommunications, electrical power systems, gas and oil storage and transportation, banking and finance, transportation, water supply systems, emergency services (including medical, fire, and rescue), etc.

Interdependent networks include identifiable industries, institutions (including people and procedures), and distribution capabilities that provide a reliable flow of products and services is one the first priorities in the governmental agendas and policy makers. In principle, these systems may produce homogeneous (e.g. only voltage) or heterogeneous (e.g. power and information flow) measurements. The trend suggests that the size of these networks is increasing in order to facilitate information gathering regarding the monitoring environment and satisfy the overall service demand. However, the increased size raises the complexity of the overall network and burdens real-time data processing. On top of that, not rarely, CIs suffer from various kinds of faults (component malfunctions, drifts, communication faults, power loss, etc.), which affect the performance of the system in a direct way. In such cases, prompt detection and isolation are of paramount importance towards avoiding information loss and/or misinterpretation of the ongoing situation.

In addition CIs may be targets of attacks (either direct or remote) aiming to disrupt their smooth functionality. The consequences of an infrastructure failing may not affect only the specific infrastructure while it has societal, health, and economic impact. Attacks on the cyber part of a Cyber-Physical (C-P) system can produce effects ranging from sporadic disruptions of field devices (sensors and actuators) to large scale outages or even loss of control in the case of a compromised industrial control system or an extended Distributed Denial-of-Service (DDoS) attack [1,2].

This work is concentrated on the automatic processing of datastreams coming from interdependent infrastructures with emphasis on the analysis of malicious events. The particular problematic is close to the scientific area of Fault Detection and Isolation (FDI), or simpler, fault diagnosis. It typically includes the detection of the fault (which refers to the time instant which the fault occurred) and its isolation (which refers to the location of the occurred fault). Fault identification corresponds to determining the nature of the detected and isolated fault, and is quite significant since it may provide useful information for designing

* Corresponding author.
E-mail addresses: stavros.ntalampiras@jrc.ec.europa.eu (S. Ntalampiras), yannis.soupionis@jrc.ec.europa.eu (Y. Soupionis), georgios.giannopoulos@jrc.ec.europa.eu (G. Giannopoulos).

http://dx.doi.org/10.1016/j.ress.2015.01.024
0951-8320/© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

a proper accommodation strategy to minimize or even eliminate the consequences of the fault. The link of fault identification has not been explored so extensively as the other links of the fault diagnosis processing chain, such as fault detection, isolation and accommodation/reconfiguration [3,4]. Identification follows detection and typically constitutes a selection of a specific kind of fault f_i out of an a-priori known set of faults $F=\{f_1,f_2,\ldots,f_Z\}$, where Z is the total number of fault types. Selection is made based on the observation of a specific symptom(s) or a sequence of them, while the classifier learns to associate them with a fault $f_i \in F$.

This article proposes a methodology for identifying malicious events on CIs without the need of an analytical model while considering the cases of an erroneous fault detection. To this end the overall network state is captured by means of a correlation map. The method is an extension of the modeling part of [5] while the approach presented here exploits the probabilistic space. We model the relationships between the datastreams coming from a CI using a hidden Markov model (HMM) trained on the parameters of linear time invariant (LTI) models estimating the relationships. Subsequently the faulty data are automatically annotated based on distance on the probabilistic space between the likelihoods observed during training and the ones computed online. The probability is a metric showing how probable it is that the specific data sequence was generated by the particular HMM. The rationale behind the usage of our approach comes from the fact that an HMM operating on the LTI space is able to address the nonlinearities existing within the dataset. Concurrently the system is able to understand whether there is a bias in the model since it relies on likelihoods based on a group of models. Overall our approach can identify whether the data belong to the fault-free situation or a malicious one while the emphasis is placed on cyber attacks.

The main aspect of our attack scenarios is that a malicious user initiates a cyber attack against the ICT network by limiting the communication/network bandwidth. One of our main goals is to monitor and measure the outcomes in a more realistic aspect than the total simulated by including the emulation of the cyber part. Unlike [6] this work focuses on the applicability of the HMM technique on an experimental test-bed composed of a simulated and an emulated environment.

The rest of this article is organized as follows: Section 2 provides an analysis of the fault identification literature focused on CIs. Next, Section 3 describes the joint usage of LTI and HMM for modeling the relationship between two datastreams and Section 4 explains the algorithm for identifying malicious states. Section 5 describes the emulator platform which was used in our experiments. In Section 6 we explain the experimental set-up, the scenarios and the obtained results. Finally the last section includes the conclusions of this work.

2. Related literature

The fault identification component is without a doubt of high importance for accommodating effectively the consequences of a potential fault. However it is not so well explored with respect to other Fault Diagnosis System's (FDS) components. Most approaches are based on an analytical mathematical model which characterizes the process under monitoring [7]. Thus they are subject to the accuracy of this model, and in the case of complex systems working under adverse real-world conditions it is not only complicated but sometimes non-realistic to derive a reliable model.

Computational intelligence methods [4] can be employed in order to overcome this obstacle. These methods can be based on quantitative (numerical) and/or qualitative (symbolic) information about the process of interest. Qualitative information is used in [8] where a fault-tree analysis was designed as an analytical troubleshooting tool by a team of knowledgeable managers, engineers, and technicians. Fault tree analysis is also used by Crosetti [9] with a probability evaluation scheme. Fuzzy if-then relations have also been used in the fault diagnosis domain. Dexter [10] created fuzzy reference models to describe the symptoms of both faulty and fault-free plant operation and subsequently used them to identify whether the system is operating correctly or a particular fault is present.

Even though qualitative computational intelligent approaches are effective, the derivation of accurate rules and/or fuzzy if-then relations is difficult, not to mention time-consuming and costly in case domain experts are involved. This makes them impractical for many engineering applications. Thus methods which can learn these rules “hidden” within large datasets are employed with neural networks constituting the primary tool due to their universal non-linear function approximation property [11]. Neural networks can model the behavior of a given system based on its produced input-output data. A work which employs NNs is reported in [12] where both artificial and real-world data were used to train NN agents for classifying between different motor bearing faults through the measurement and interpretation of motor bearing vibration signatures. Fault diagnosis in non-linear dynamic systems based on neural networks is described in [13]. This work uses a multi-layer perceptron network trained to predict the future system states based on the current system inputs and states. Afterwards, a neural network is trained to classify characteristics contained in the residuals and essentially perform fault identification.

Several works in the literature aim at exploiting the merits of both qualitative and quantitative approaches. Yu et al. [14] exploits analytical redundancy via parity equation while neural networks are then used to maximize the signal-to-noise ratio of the residual and to isolate different faults. This methodology is applied for fault detection and isolation for a hydraulic test rig. Ming et al. [15] proposes the usage of multilevel flow models and ANN to develop a fault diagnosis system, with the intention of improving both identification and understandability of the diagnostic process and results. A feedforward ANN trained with the BP algorithm is employed and when the faults are localized a diagnosis is performed by ANNs for either confirming the faults or offer an alternative solution and/or detailed information about the possible root cause. The application scenario is a Nuclear Power Plants simulator.

2.1. The case of critical infrastructures

The literature includes a variety of solutions for fault detection in infrastructures. However these are of limited-scope and address a relatively narrow part of the problem space. Shames et al. [16] proposes the usage of a bank of observers via a model-based fault diagnosis method, where a set of residuals is generated indicating the presence of a fault. A model is created which explains the nominal state of the system while the detector evaluates the discrepancy between its estimates and the actual measurements. A fault is detected when the discrepancy is over a threshold. The authors employed artificial data to provide illustrative examples of how their methodology may be applied on power networks which can be thought as very complex infrastructures in which generators and loads are dynamically interconnected. Villez et al. [17] provides a basis for the integration of diverse Fault Detection and Isolation (FDI) methods as well as optimal coupling of FDI and control modules in the closed-loop supervisory control system. Furthermore it gives the expected impact and perspectives of the proposed Bayesian fusion schema without reporting experimental results. The authors of [18] investigate the interactions existing between infrastructure systems which may lead to increased or decreased risk of failure in each individual system. To this end they employ a set of diverse models for formulating appropriate risk assessment tools. A power transmission grid is used as an example focusing on blackout dynamics, while data is generated both by a probabilistic and a dynamical model.

Bovenzi et al. [19] proposes an anomaly-based approach for the online detection of faults based on Statistical Predictor and Safety Margin (SPS). The proposed SPS anomaly detection algorithm has been applied on the Air Traffic Management domain. Their algorithm is compared with a relatively simple one which adopts static thresholds on a dataset coming from the SWIM-BOX® [1] which includes various types of parameters such as syscall errors, signals, and scheduling time of processes. An interesting work is presented in [20] where the emphasis is placed on identifying and discriminating attacks and faults under the scope of improving the security of Electric Power Control Systems. The methodology is based on a set of rules discovered by using the Rough Sets Classification Algorithm. The authors intent to experiment using the network model IEEE RTS-96 [21] in an electric power simulator. A fuzzy-neural data fusion system aiming at increased state-awareness of resilient control systems is reported in [22]. A data fusion mechanism is associated with each component of the control system. Furthermore, each mechanism is composed three layers (*a*) conventional threshold-based alarms, (*b*) anomalous behavior detector using self-organizing maps, and (*c*) prediction error based alarms using neural network based signal forecasting. Experimental results are derived using data coming from a Matlab Simulink model of the Idaho National Laboratory Hytest process, which is a testing facility for hybrid energy systems composed of tightly-coupled chemical processes [23].

A quite interesting as well as new direction is presented in [24] which explains the development of a terrorist attack prediction model using dynamic Bayesian networks. The network is able to provide a likelihood of future terrorist activities at critical transportation infrastructure facilities. However only its theoretical development is presented and its potential effectiveness is demonstrated by two examples where the aim is to predict a terrorist strike with the possibility of an airplane hijack at a typical U.S. airport.

An SVM based anomaly detection algorithm is described in [25]. It uses the Kolmogorov-Smirnov test for automatically estimating the SVM parameters. The authors conducted experiments on two telecommunication network data sets for detecting *undesired* events while their evaluation is solely qualitative.

The same problematic from another point of view is given in [26] which presents two approaches for gaining knowledge which might be proven useful when designing critical infrastructures, while the problem is put under the umbrella of reliability and vulnerability analysis. Nan et al. [27] explains the interdependencies between two exemplary systems (SCADA and SUC) using a modified five-step methodical framework. Based on this analysis, the paper suggests methods for improving the system performance. The article [28] provides a comprehensive description of the interactions between public policy, managerial decision-making and the engineering of critical infrastructures while [29] analyses the background and recent applications of stochastic point processes in reliability analysis.

To the best of our knowledge there is no approach in the literature using the probability space of a group of HMMs for assessing the state of interdependent critical infrastructures. Most approaches in the literature are qualitative and based on analytical models of the underlying process. With an HMM operating on the parameter space we are able to work on generic datastreams for capturing data redundancies while addressing potential nonlinearities. In addition we propose the usage of a correlation map giving us the ability to assess the network state using data coming from fewer nodes. Furthermore our experiments are not simulated but we employed a powerful emulation platform (Section 5). The overall aim is to provide recommendations on how monitoring strategies should be deployed in order to contribute towards increasing the overall resilience of the network.

3. Modeling the relationships between datastreams

This section explains the method used for modeling the relationships between datastreams coming from correlated sensors, meaning that the pattern of the relationship should remain consistent when the system operates in a certain state (faulty or not).

Let us consider a monitoring framework comprised of K node sensors, each of which generates a datastream. Denote by $X_i : \mathbb{N} \rightarrow \mathbb{R}$ the datastream acquired by the ith sensor. Table 2 includes the symbols used in the present analysis.

Let $O_{i,T_0} = \{X_i(t), t = 1, \ldots, T_0\}$ and $O_{j,T_0} = \{X_j(t), t = 1, \ldots, T_0\}$ be the data sequence of the ith and jth sensors. In the following we assume that their relationship is characterized by a process $\mathcal{P}$ which is time-invariant or that every state of the system (e.g. nominal, fault, attack, etc.) can be approximated by a sequence of models even if it is time-variant (e.g. through a Markov process in the parameter space).

Therefore, to construct a model for their relationship, we consider the general *discrete-time linear MISO* structure [30]:

$$A(z)X_i(t) = \sum_{j=1}^{m} \frac{B(z)}{F(z)} X_j(t) + \frac{C(z)}{D(z)} d(t), \tag{1}$$

where $d(t)$ is an independent and identically distributed random variable accounting for the noise, m is the number of inputs, z is the time-shift operator while $A(z), B(z), C(z), D(z)$ and $F(z)$ represent z-transfer functions, whose parameter vectors are $\theta_A, \theta_B, \theta_C, \theta_D$ and θ_F respectively. Consequently an element f_θ in the approximating model family $\mathcal{M}(\theta)$ is fully described with a $\theta \in \mathbb{R}^p$ which comprises the above parameter vectors. Following the logic of [31], we create an ensemble of dynamic models (e.g. ARX, ARMAX, and OE) with various orders and select the one which best fits the datastreams (i.e. lowest reconstruction error) while low-order models are preferred. The model search algorithm minimizes a robustified quadratic prediction error criterion. It should be mentioned that the methodology is independent of the selected model type and can be applied unaltered.

The utilization of linear models ensures that the regularity assumptions imposed by [30,32] are satisfied. Thus, our framework is placed on a solid mathematical background despite the introduced model bias $\| f_\theta - \mathcal{P} \|$ suggesting that the underlying distribution of the parameters is a multivariate Gaussian (the bias here is seen as a time-invariant "difference" between the predicted and the true process). However various models are needed to describe a specific source of data, the number and the connections of which is not known a priori. A hidden Markov model is appropriate for dealing with this type of bias since it can break the problem into a specific number of states which are connected in a probabilistic way (Fig. 1).

We model the sequence of the model parameters by means of an HMM:

$$H_{\theta_T} = \{N, P, A, \pi\}, \tag{2}$$

where N are the states, P is the probability density functions with respect to each state, θ_T are the parameters of the training sequence, A is the state transition probability matrix and $\pi$is the initial state distribution. The dynamic model space is searched based on the log-likelihood criterion during the operational life. The model which produces the highest log-likelihood is selected out of the library which is created off-line. The following

[1] http://www.swim-suit.aero/swimsuit/.

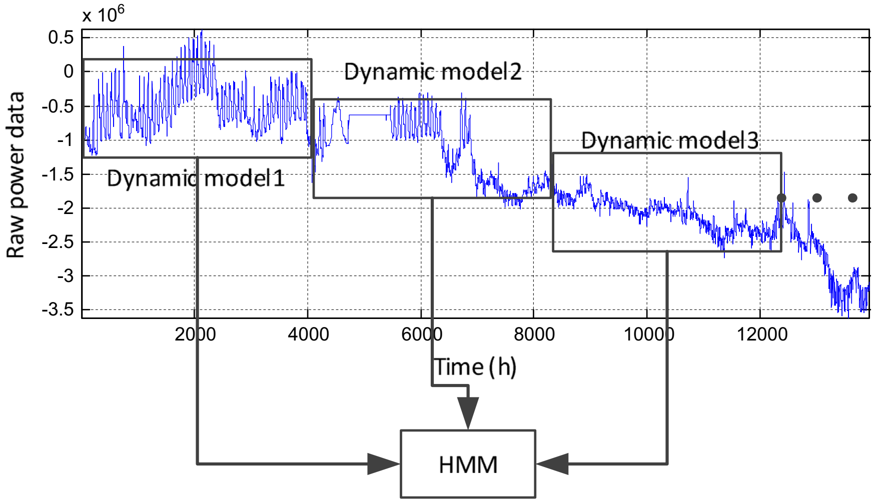

Fig. 1. A demonstration of the way the HMM operates and addresses the model bias inserted during the linear time invariant modeling phase.

subsections explain the usage of the HMM modeling approach in the context of fault diagnosis.

3.1. Hidden Markov models

Hidden Markov models constitute an extension of the discrete Markov processes while the main focus is placed on real-world problems. HMMs have been proposed in [33], where the observation is a probabilistic function of the state. The resulting model includes two stochastic processes, one of which is not observable (hidden) and can only be observed through another set of stochastic processes which produce the sequence of observations. In an HMM the states are referred to as hidden because the system we wish to model may have underlying causes that cannot be observed.

An HMM is characterized by the following components:

- the number of states N;
- the probability density function associated with each state modelled as a mixture of Gaussians (GMM), $P(x|\theta) = \sum_{k=1}^{K} p_k p(x|\theta_{(k)})$, where p_k's are the mixture weights, x is a continuous-valued data vector (e.g. measurements or features), $\theta_{(k)}$ represents the kth component of the vector, $\theta = [\sigma, \mu]$, $p(x|\theta_{(k)}) = 1/(2\pi)^{d/2}|\sigma_k|e^{-1/2(x-\mu_k)^t \sigma_k^{-1}(x-\mu_k)}$;
- the state transition probability matrix $A = \{a_{ij}\}$ where entry a_{ij} represents the probability of moving from state j at time t to state i at time $t+1$. For the case where the system may transit to any state at a given time instant, we have $a_{ij} > 0, \forall i, j$. In case some transitions are not allowed, the respective a_{ij}s should be set to zero;
- the initial state distribution $\pi = \{\overline{\pi_i}\}$, where $\overline{\pi_i}$ corresponds to the probability that the HMM starts in state i, i.e. $\pi_i = p[S_i], 1 \le i \le N$.

3.2. HMM Training

Model parameters, that is, *the transition probabilities, emission probabilities and the initial state probability* need to be adjusted so as to maximize the probability of the observed sequence and adequately represent the training set. The Baum–Welch algorithm [34] is a method that uses an iterative approach and provides a solution to this problem. It starts with preassigned probabilities and tries to adjust them based on the observed sequences in the training dataset.

The HMM parameters can be initialized to predetermined values or to a constant before applying the Baum–Welch algorithm. As the path taken is not known, the algorithm counts the number of times each component is used when the observed set of elements in the training sequence is given to the present HMM. Each iteration of the algorithm includes two steps, the Expectation step (E Step) and the Maximization step (M Step). The Maximization step uses the counts of the number of times an element is seen at a state and the number of times a transition occurs between two states which were obtained from the Expectation step to update the transition and emission probabilities in order to maximize the performance. The algorithm stops when the convergence criterion is satisfied (the log-likelihood between subsequent iterations is under a threshold, $|L_{t+1} - L_t| < T_{Viterbi}$) or when the maximum number of permitted iterations is reached.

3.3. Log-likelihood computation of unknown data

The Viterbi algorithm is used to find the most probable path taken across the states in the HMM. The algorithm checks all possible paths leading to a state and gives the most probable based on dynamic programming. It keeps track of the best state used during a transition using pointers. The most probable path is found by moving through the pointers backwards starting from the end state to the start state. Sometimes we may obtain more than one path as the most probable; in such cases one path is randomly selected. The Viterbi Algorithm is analytically explained in [35].

3.4. The dependency graph

While datastreams coming from each element of the power network are all correlated to some extent we need to reduce the number of functional dependencies to the most relevant ones for reducing the computational cost. In fact, poorly correlated datastreams reflect a weak functional dependency yielding to poor performing models which may decrease the overall performance of the FDS. The success of this approach may define also the applicability of the concept to detect issues in the network with only a small subset of information coming from a reduced number of nodes.

The first stage of this preliminary step is the estimation of the cross-correlations among the datastreams coming from all the K power nodes. Afterwards, a dependency graph, which initially comprises all the possible relationships among the nodes, is pruned by discarding those relations whose cross-correlation peak is below a user-defined threshold. The result is a reduced dependency Fig. 2 shows the reduced dependency graph of the IEEE 30 bus system considered in the experimental section. For visibility reasons we included data coming from six buses (1, 11, 18, 21 and 25). Arcs

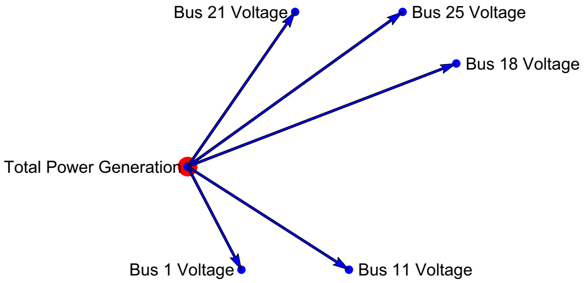

Fig. 2. The reduced dependency graph depicts the existing relevant relations among the elements of the power network. Only relationships with correlation above a certain threshold may be considered (here Threshold=0.2).

connecting CI elements represent highly-correlated functional dependencies.

After early experimentations we decided to employ bus voltages since they provide quite useful information for understanding the state of the network. In the proposed methodology the likelihood computed for bus i is weighted so that the decision upon its datastream is a function of the statuses of the buses belonging to the rest of the network. More precisely each bus is associated with two likelihoods: L_i is the one produced by HMM i and $L_{r,i}$ is a weighted sum of the likelihoods of the buses representing the rest of the network: $L_{r,i} = \sum_{j=1, j \neq i}^{30} w_{i,j} \times L_j$, where $w_{i,j}$ is the cross correlation among buses i and j and L_j the likelihood produced by the HMM associated with bus j. This way the algorithm is able to assess the status of the CI network even when information from one bus is missing or is identified as compromised. In addition a bias affecting a statistical model does not impact the final decision.

Algorithm 1. The fault diagnosis algorithm which models the relationship between two datastreams by means of an HMM.

```
1. Build the HMM representing the nominal class, $H_N = \{S_N, P_N, A_N, \pi_N\}$ from the vectors of parameters $\theta_1 \ldots \theta_d$ each of which associated with a linear dynamic model applied to the training data $O_{i,T_0, i=1,\ldots,d}$ windowized using length $M$ overlapping by $M-1$;
2 Windowize the incoming novel data as above, which results in windows $W = W_1 \ldots W_x$;
repeat
    3. $j = 1$;
    4. Compute the parameter vectors of the $j-$th dynamic model $\theta_j$ with respect to $W_j$;
    5. Compute the $\log - $ likelihood $L_{W_j} = P(\theta_1 \ldots \theta_j | H_{f_N})$;
    6. Compute the $\log - $ likelihood $L_r = \sum_{k=1, k \neq N}^{30} w_k \times L_k$;
    7. Compute $\mathcal{D} = (L_{W_j} + L_{r,W_j} - L_{Tr}), L_{Tr} = \{L_N, L_f, L_{dos}, L_{int}\}$;
    8. Find $\min(\mathcal{D})$ and assign to $W_j$ the class with the lowest discrepancy;
    9. $j = j+1$;
until FDS turned OFF;
```

4. The fault identification algorithm

The training phase of the proposed methodology creates one HMM per existing functional relationship within the network under monitoring. During testing one examines the probability generated by the created HMMs. Finally the system computes the discrepancy between the probabilities collected during the training phase and the ones produced during the testing one. The class with the lowest discrepancy is assigned to the unknown data. Based on the specific logic we essentially try to quantify the statistical similarity between the unknown data and the one available during training. The proposed approach is motivated by the fact that faults, denial of service attacks and integrity threats are associated with network data exhibiting different patterns, thus belonging to diverse probabilistic spaces.

The fault identification algorithm is summarized in Algorithm 1. We assume a training set corresponding to $O_{i,T_0, 1 \le i \le N}$ associated with each normal data. We compute the d model coefficients over a predefined window of the sensor measurements of size M. They are used to train the HMM which is to characterize the nominal class. In order to identify the HMMs with the best classification capabilities, we build a variety of HMMs with different parameters (number of states and Gaussian components) and we select the HMM based on the highest recognition rate criterion.

When unknown data is processed, it is first windowized and the model coefficients with respect to each window are computed and inserted into the trained HMM. The log-likelihood is then calculated for window W_j and its identity is determined by computing its difference from the log-likelihoods seen during training. The classification process is shown Fig. 3.

5. Experimental framework and implemented attacks

As an experimental framework we have used AMICI, a novel Assessment platform for Multiple Interdependent Critical Infrastructures [36]. AMICI uses simulation for the physical components and the emulation testbed, EPIC, based on Emulab [37,38] in order to recreate the cyber part of CIs, e.g., BGP routing protocols, SCADA (Supervisory Control And Data Acquisition) servers, corporate network. The use of simulation for the physical layer is a very reasonable approach due to small costs, the existence of accurate models and the ability to conduct experiments in a safe environment. The argument for using emulation for the cyber components is that the study of the security and resilience of computer networks would require the simulation of all the failure related functions, most of which are unknown in principle.

Whenever real-time simulation is used, models run in a discrete time-domain that is closely linked to the clock of the OS. This means that the simulated model runs at the same rate as the actual physical system. We use generic PCs with multitasking OSs to run the real-time software simulation units. Our choice to use Simulink Coder based on well-established Matlab-based software as Matpower [39] and Matdyn [40] to produce the simulators. An important aspect in this sense is the choice of the model

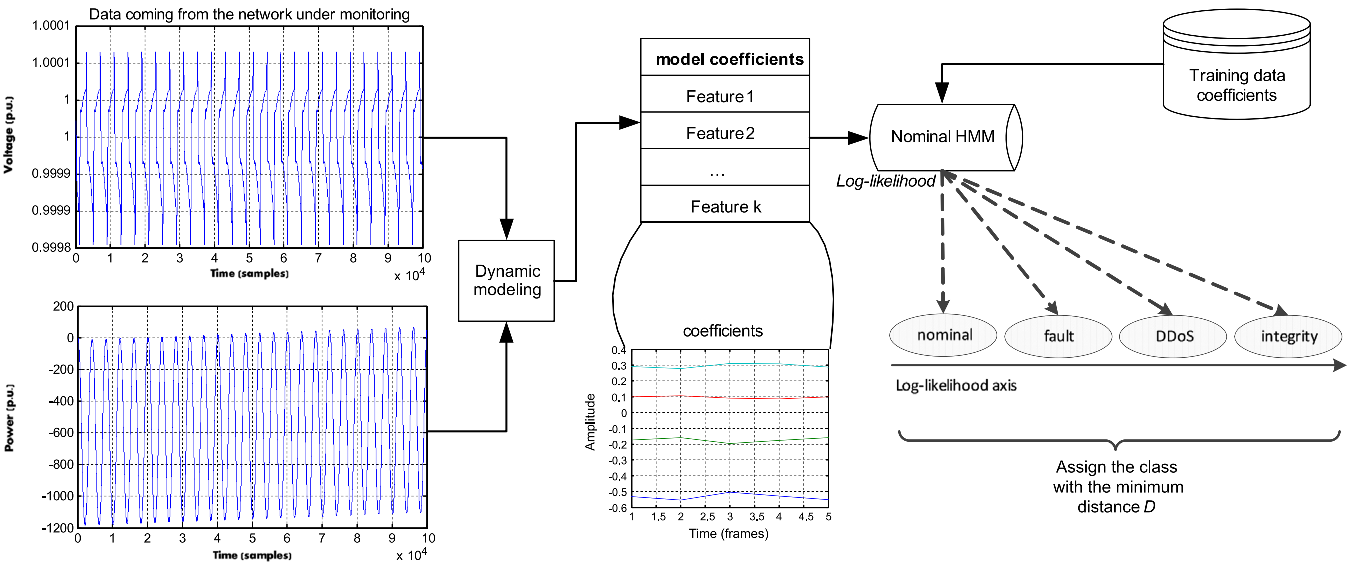

Fig. 3. Fault identification based on the nominal HMM where the class of the unknown data is determined by the distance in the log-likelihood space.

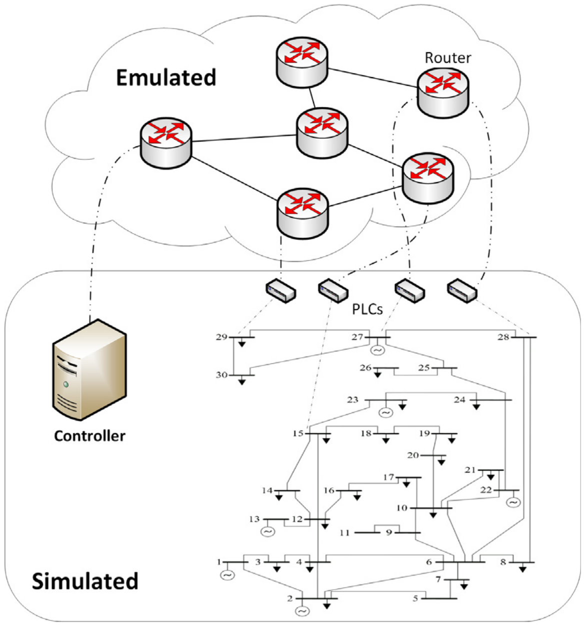

Fig. 4. The experimentation architecture.

execution rate, also known as the simulation step, where we verified that the output of real-time simulation reproduces as accurately as possible the real-world process.

From the simulation point of view each model is seen as a set of inputs and outputs. These are mapped to an internal memory region (I/O MEM) that is read/written by other software modules as well. This way, AMICI enables the implementation of Programable Logic Controllers (PLCs) and the interaction with remotely software model by mapping messages from Remote Procedure Call (RPC) messages to Modbus and vice-versa , i.e., can exchange model values as measured voltage. Interaction with other simulation units is enabled by implementing not only RPC server-side operations but client-side calls as well. These data is exchange through our emulated network which brings our implementation as close as possible to real infrastructures.

5.1. Cyber-attacks

We classify attacks on the cyber-part of C-P systems into integrity attacks and Distributed Denial of Service (DDoS) attacks . The above classification is based on the resources available to the attacker, skill level and his expertise in C-P system operation and control. If the attacker has knowledge a priori on the operation of the entire system, he would be able to produce a much greater threat. Thereby, with this knowledge, an attacker would be in a better position to compromise a substation that is very vital to the power infrastructure.

Distributed Denial of Service (DDoS) attacks are one of the most effective attacks that modern CIs must cope with. Massively distributed DDoS attacks rely on thousands of infected hosts to flood the victim with a large number of packets and consume its resources, mostly communications bandwidth. Consequently, the victim is left without communications resources and it effectively looses control over remotely controlled installations. Moreover, the response strategy for the controller during a DDoS attack is the last received data/signal to be treated as current command:

$$c(t) = \begin{cases} c^{past}(t) & \in T_{DoS} \\ c^{real}(t) & \notin T_{DoS} \end{cases} \tag{3}$$

Our focus on the integrity attacks is placed on *replay attacks*. All messages contain time-varying information which reflects the current system status and actions required. Attackers can catch some messages and deliver them afterwards. This kind of attacks can also lead power grid control to malfunctions. Finally, if the attacker has initiated the data recording at t_1 and finished at t_2, then the reply attack can be illustrated as

$$c(t) = c(t'), \text{where} \quad t_1 < = t' < = t_2 \quad \& \quad t > t_2 \tag{4}$$

6. Experiments

The experimental section provides details regarding (*a*) the simulation and emulation environment used in the current study, (*b*) the dataset coming from the experimental test-bed, and (*c*) the parametrization of the proposed approach as well as its performance analysis.

6.1. The simulation and emulation environment

The used implementation framework is illustrated fig 4 which shows (a) simulated the physical components of a power grid and (b) emulated the cyber elements. The power grid integrated in this experiment is the well-known IEEE 30-bus model (see Fig. 1 for its graphical representation). It includes (a) 6 generator buses, (b) 5

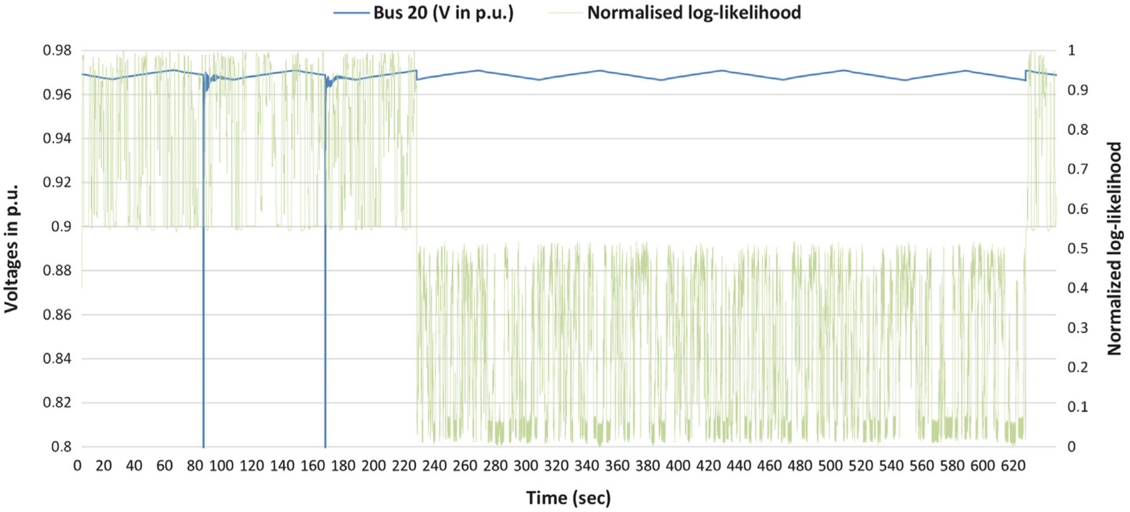

Fig. 5. The operation of the proposed algorithm during an integrity attack. In the beginning the bus voltage fluctuates due to two faults occurring on neighbor buses while the integrity attack starts at the 220th second and causes a drop in the probability of the HMM. The attack is detected with a delay of 6 samples.

connecting buses, and (c) 20 buses where the consumers are connected buses. A few busses can simultaneously have a generator and a consumer connected to them. Moreover, the sampling rate (simulation step) of the simulated model is 21ms which it the time to calculate its parameters including the interaction with the emulated environment.

The cyber-part includes (a) 6 routers (Cisco 6503), which have four Gigabit experimental interfaces and one control interface (emulation) and (b) 6 virtual PCs (HP Proliant GL380p), which have Xeon(R) 4 CPUs@2.40 GHz, 3GB RAM, two Gigabit experimental interfaces and one control interface. They were used as experimental nodes (attackers and simulated elements) and their operating system is FreeBSD8.2.

Finally, a network measurement reveals an average Round Trip Time (RTT) below 3 ms. This means that the implementation exhibits the operational behavior of real communications systems where the delivery of high-speed messages must be below the maximum limit of 10ms, as stated by the IEEE 1646-2004 standard [41] on communications delays in substation automation.

6.2. The training scenarios

The detection method presented in Section 4 needs a training process before being able to provide real-time results. The data from the training scenario helps as an input to establish the legitimate and malicious states of the exchanging data from the PLCs to controller. The experimental scenario sets the normal state of the power grid. Limited fluctuations in consumed load are present without faults and/or attacks.

6.3. FDS parameterization

The HMMs have been configured in a fully connected topology (ergodic HMM), which means that the algorithm permits every possible transition across states. Lastly, the distribution of each state is modeled by a GMM with a diagonal covariance matrix.

We employed the Torch framework[2] during both learning and validation phase. The maximum number of k-means iterations for cluster initialization was set to 50 while the Baum–Welch algorithm used to estimate the transition matrix was bounded to 100 iterations with a threshold of 0.001 between subsequent iterations. The number of explored states ranges from 3 to 10 while the number of Gaussian components used to build each GMM comes from the {2, 4, 8, 16, 32, 64, 128, 256 and 512} set. We did not apply any data pre-processing techniques, such as normalization, to avoid any kind of information loss. The particular parameters refer on both single and universal HMM modeling. Finally the window length M was 100, a value which provided satisfactory reconstruction error during the preliminary experimental phase.

The correlation threshold (see Fig. 2), was chosen via a sensitivity analysis focusing on the fact that there should not be isolated parts in the network. A malicious event including attacks, faults, etc. may appear at any infrastructure node, thus data coming from all the nodes should be collected and processed. However their significance alters according to the weights of w_j of the correlation map.

We carried out a thorough experimental procedure to evaluate the effectiveness of the proposed method. The following figures of merit were used [5]:

- False positive rate (FP): it counts the number of times the algorithm detects a malicious situation in the datastream when there is not.
- False negative rate (FN): it counts the number of times the algorithm does not detect a malicious situation in the datastream even though there is.
- Detection delay (DD): it measures the delay in number of samples for detecting a malicious situation.

The main attack scenario was a replay attack (Section 5) against the data transmitted from BUS 20 to the controller (see Fig. 5). We used the tcpdump[3] tool for recording the data traffic and tcpreplay[4] tool for replaying previously captured network traffic. During this attack we executed faults and DDoS attacks against other buses in order to see how our method work when the provided information is limited. The DDoS attack was against a router within a given period of time where most/all of the available bandwidth was consumed. Therefore partially or none data reached the controller. The DDoS attack was implemented by using the iperf tool.[5] In this case the algorithm discards the attacked bus and infers the state of the system using information coming from the rest of the network.

Each attack had a duration of more than 30,000 samples (the sampling rate of the simulated model is 20 ms). The fault diagnosis

[2] Torch Machine Learning Library, available at http://www.torch.ch/.

[3] A command line network sniffer, available at http://www.tcpdump.org/.

[4] Utility for editing and replaying previously captured network traffic, available at http://tcpreplay.appneta.com/.

[5] Tool to measure network performance, available at http://iperf.fr/.

framework is trained on data coming from the normal modality of total length of 15,000 samples while the rest were used for testing. The system models the overall power and the load of each bus. Model training required 14 days on a PC running Windows 7 with double core processor and 32 GB of RAM while testing was done in real time since only computations of log-likelihoods are included which are computationally inexpensive.

The model order is determined by minimizing a quadratic prediction error criterion. The model family providing the lowest error was ARX while the model fits the data in real time. Finally, the one provided the best performances was the following ARX (2,2):

$$X_i(t) = a_1 X_i(t-1) + a_2 X_i(t-2) + b_1 X_j(t-1) + b_2 X_j(t-2) \tag{5}$$

where $a_1 = 0.5$, $a_2 = 0.2$, $b_1 = 0.1$, $b_2 = 0.3$. It should be noted that for a different network another model may be proven optimal by the process explained in Section 3. The figures of merit are computed both on data coming from the nominal and malicious states. They are tabulated in Table 1.

Table 1
The detection results of the proposed and the parity equation methods in the following format: proposed/parity. The figures of merit are averaged over the entire dataset.

Test data type	FP (%)	FN (%)	Detection delay (# of samples)
Nominal	0/13.4	–/–	–/–
Overload (fault-free)	0/19.1	–/–	–/–
Underload (fault-free)	0/15.9	–/–	–/–
Fault	–/–	0.5/12.7	8.6/57.2
DDoS	–/–	0.2/8.5	2.1/17.2
Integrity	–/–	1.1/17.5	2.5/62.7

6.4. *Experimental results*

Since the literature does not include an approach for fault identification in interdependent CIs, the method proposed here was contrasted to the parity equation approach which represents well the literature on fault detection [7,42]. Here we observe the discrepancy between the process behavior and the process model describing the nominal fault-free behavior. The threshold for detecting a fault is set equal to the highest discrepancy on unseen training data. In case the discrepancy surpasses the threshold, a faulty situation is detected.

During our experimentations we used data coming from all the nodes of the electrical network while all the malicious states were correctly recognized. It should be mentioned that we used the same training and testing sequences among the different methods in order to achieve a fair comparison. In addition Table 1 provides the FPs, FNs and detection delays with respect to every type of network state for both methods. We can observe that the proposed method offers superior figures of merits for all system states including the nominal and the faulty ones. The metrics offered by the proposed method are within quite low limits while the most difficult to detect are the integrity attacks (highest FP, FN and delay). Conclusively the overall performance of the system is quite promising especially when considering the degree of complexity of the problem as well as the incorporation of two diverse and interdependent CIs.

7. Conclusions

This work explained and thoroughly evaluated a novel framework for identifying malicious situations appearing in the context of interdepended CIs. The experimental set-up consisted of a simulated IEEE 30 bus energy network operated via an emulated

Table 2
List of symbols.

X_i	the stream of data acquired by the i-th sensor
O_{i,T_0}	the data sequence of the i-th sensor for $t = 1, \ldots, T_0$
$\mathcal{P}$	time-invariant process
$d(t)$	independent and identically distributed random variable for noise
z	time-shift operator
$A(z), B(z), C(z), D(z)$ and $F(z)$	z-transfer functions, whose parameter vectors are $\theta_A, \theta_B, \theta_C, \theta_D$ and θ_F respectively
$\mathcal{M}(\theta)$	model family with parameters θ
H_{θ_T}	HMM trained on parameters θ coming from the interval $t=1,\ldots,T$
N	number of states
P	probability density function
A	state transition probability matrix
π	initial state distribution
p_k	mixture weight
σ	variance of a Gaussian
μ	mean of a Gaussian
a_{ij}	the probability of moving from state j at time t to state i at time $t+1$
L_i	likelihood of HMM i
$L_{r,i}$	weighted sum of the likelihoods of the buses excluding bus i
$w_{i,j}$	the cross correlation between a bus i and bus j
K	number of sensors
$\mathcal{D}$	distance measured in the probabilistic space
M	data window size
W	data window
d	number of LTI model coefficients
$c(t)$	data value at time t
Z	number of fault types
m	number of inputs
f_i	fault of type i, $i \in \mathcal{Z}, i \in \{1, \ldots, Z\}$
k	Gaussian component

telecommunication network. The proposed method models the sequence of the parameters of LTI models by means of an HMM, while the decision regarding the state of the system of CIs (nominal, under DDos or integrity attack) is made using a distance metric in the probabilistic space. An interesting feature of the algorithm is its ability to provide a system state even when a bus is down or under attack using a correlation map of the network. After extensive experimentations, it was shown that the performance of the system is encouraging as measured using false positive rate, false negative rate and detection delay.

We aim to extend this work by cooperating with a smart grid operator and apply it on real-world data. In addition we aim to generalize the method by designing a scheme for detecting data which are not statistically similar to the ones seen during training and thus may comprise new kinds of malicious situations. The similarity could be based on a statistical distance metric, e.g. Kullback–Leibler divergence and when an adequate amount of new data is gathered, a new class of malicious events can be determined. We believe that such a module would be a great help while the operators need to analyze a malicious event and deduct the vulnerabilities of their infrastructure which lead to it. Another interesting direction is the development of a mechanism for correcting the decisions made by the fault identification algorithm based on the ordering in time, i.e. faults of unreasonable duration or improbable series of faults. Finally we wish to study the occurrence of multiple faults at the same time by evaluating a number of hypotheses equal to every possible combination of fault classes.

References

[1] Wang W, Lu Z. Cyber security in the smart grid: survey and challenges. Comput Netw 2013;57(5):1344–71. http://dx.doi.org/10.1016/j.comnet.2012.12.017. ⟨http://www.sciencedirect.com/science/article/pii/S1389128613000042⟩.

[2] Yan Y, Qian Y, Sharif H, Tipper D. A survey on cyber security for smart grid communications. IEEE Commun Surv Tutor 2012;14(4):998–1010.

[3] Hwang I, Kim S, Kim Y, Seah CE. A survey of fault detection, isolation, and reconfiguration methods. IEEE Trans Control Syst Technol 2010;18(3):636–53. http://dx.doi.org/10.1109/TCST.2009.2026285.

[4] Sobahni-Tehrani E. Fault detection, isolation, and identification for nonlinear systems using a hybrid approach. Canadian theses, Concordia University, Canada; 2008. ⟨http://books.google.it/books?id=zCPbPiEo15gC⟩.

[5] Alippi C, Ntalampiras S, Roveri M. A cognitive fault diagnosis system for distributed sensor networks, IEEE Trans Neural Netw Learn Syst http://dx.doi.org/10.1109/TNNLS.2013.2253491.

[6] Pasqualetti F, Dorfler F, Bullo F. Attack detection and identification in cyber-physical systems. IEEE Trans Autom Control 2013;58(11):2715–29. http://dx.doi.org/10.1109/TAC.2013.2266831.

[7] Isermann R. Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Verlag; 2006.

[8] Zampino E, Application of fault-tree analysis to troubleshooting the NASA GRC icing research tunnel, in: Proceedings of the 2001 annual reliability and maintainability symposium; 2001. p. 16–22. http://dx.doi.org/10.1109/RAMS.2001.902435.

[9] Crosetti PA. Fault tree analysis with probability evaluation. IEEE Trans Nucl Sci 1971;18(1):465–71. http://dx.doi.org/10.1109/TNS.1971.4325911.

[10] Dexter A. Fuzzy model based fault diagnosis. IEE Proc Control Theory Appl 1995;142(6):545–50. http://dx.doi.org/10.1049/ip-cta:19952089.

[11] Cybenko G. Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 1989;2:303–14.

[12] Li B, Chow M-Y, Tipsuwan Y, Hung J. Neural-network-based motor rolling bearing fault diagnosis. IEEE Trans Ind Electron 2000;47(5):1060–9. http://dx.doi.org/10.1109/41.873214.

[13] Patton R, Chen J, Siew T. Fault diagnosis in nonlinear dynamic systems via neural networks. In: International conference on control '94, vol. 2; 1994. p. 1346–51. http://dx.doi.org/10.1049/cp:19940332.

[14] Yu D, Shields DN, Daley S. A hybrid fault diagnosis approach using neural networks. Neural Comput Appl 1996;4(1):21–6.

[15] Ming Y, Jianfei L, Minjun P, Shengyuan Y, Zhijian Z. A hybrid approach for fault diagnosis based on multilevel flow models and artificial neural network. In: CIMCA/IAWTIC. IEEE Computer Society; 2006. p. 2.

[16] Shames I, Teixeira AMH, Sandberg H, Johansson KH. Distributed fault detection for interconnected second-order systems with applications to power networks. In: First workshop on secure control systems. 2010.

[17] Villez K, Venkatasubramanian V, Garcia H, Rieger C, Spinner T, Rengaswamy R. Achieving resilience in critical infrastructures: a case study for a nuclear power plant cooling loop. In: 2010 3rd International Symposium on Resilient Control Systems (ISRCS). 2010. p. 49–52. http://dx.doi.org/10.1109/ISRCS.2010.5602159.

[18] Newman DE, Nkei B, Carreras BA, Dobson I, Lynch VE, Gradney P. Risk assessment in complex interacting infrastructure systems, in: Proceedings of the 38th annual Hawaii international conference on system sciences (HICSS'05)—Track 2, vol. 02. IEEE Computer Society, Washington, DC, USA; 2005. p. 63.3. http://dx.doi.org/10.1109/HICSS.2005.524.

[19] Bovenzi A, Brancati F, Russo S, Bondavalli A. A statistical anomaly-based algorithm for on-line fault detection in complex software critical systems. In: Proceedings of the 30th international conference on computer safety, reliability, and security, SAFECOMP'11, Springer-Verlag, Berlin, Heidelberg; 2011. p. 128–42. ⟨http://dl.acm.org/citation.cfm?id=2041619.2041634⟩.

[20] Coutinho M, Lambert-Torres G, da Silva L, et al. Attack and fault identification in electric power control systems: an approach to improve the security. In: Power Tech, 2007. IEEE, Lausanne; 2007. p. 103–07. http://dx.doi.org/10.1109/PCT.2007.4538300.

[21] Grigg C, Wong P, Albrecht P, et al. The IEEE reliability test system—1996. A report prepared by the reliability test system task force of the application of probability methods subcommittee. IEEE Trans Power Syst 1999;14(3):1010–20. http://dx.doi.org/10.1109/59.780914.

[22] Linda O, Manic M, McJunkin T. Anomaly detection for resilient control systems using fuzzy-neural data fusion engine, in: 2011 4th international symposium on resilient control systems (ISRCS). 2011. p. 35–41. http://dx.doi.org/10.1109/ISRCS.2011.6016085.

[23] Stoots JOC, Shun L. Integrated operation of the INL Hytest system and high-temperature steam electrolysis for synthetic natural gas production. In: Proceedings of 2nd international meeting of the safety and technology of nuclear hydrogen production, control and management. June 2010.

[24] Jha MK. Dynamic Bayesian network for predicting the likelihood of a terrorist attack at critical transportation infrastructure facilities. J Infrastruct Syst 2009;15:31. http://dx.doi.org/10.1061/(ASCE)1076-0342(2009)15:1(31) ⟨http://link.aip.org/link/?JITSE4/15/31/1⟩.

[25] Jiang Y, Jiang J, Capodieci P. A SVM-based behavior monitoring algorithm towards detection of un-desired events in critical infrastructures. In: Herrero L, Gastaldo P, Zunino R, Corchado E, editors, CISIS, vol. 63 of Advances in intelligent and soft computing. Springer; 2009. p. 61–8. ⟨http://dblp.uni-trier.de/db/conf/cisis-spain/cisis2009.htmlJiangJC09⟩.

[26] Johansson J, Hassel H, Zio E. Reliability and vulnerability analyses of critical infrastructures: comparing two approaches in the context of power systems. Reliab Eng Syst Saf 2013;120:27–38.

[27] Nan C, Eusgeld I, KrÃger W. Analyzing vulnerabilities between {SCADA} system and {SUC} due to interdependencies. Reliab Eng Syst Saf 2013;113:76–93. http://dx.doi.org/http://dx.doi.org/10.1016/j.ress.2012.12.014 ⟨http://www.sciencedirect.com/science/article/pii/S0951832013000033⟩.

[28] Johnson CW. Understanding the interaction between public policy, managerial decision-making and the engineering of critical infrastructures. Reliab Eng Syst Saf 2007;92(9):1141–54. http://dx.doi.org/http://dx.doi.org/10.1016/j.ress.2006.08.011 ⟨http://www.sciencedirect.com/science/article/pii/S0951832006001700⟩.

[29] Krivtsov VV. Recent advances in theory and applications of stochastic point process models in reliability engineering. Reliab Eng Syst Saf 2007;92(5):549–51.

[30] Ljung L. Convergence analysis of parametric identification methods. IEEE Trans Autom Control 1978;23(5):770–83. http://dx.doi.org/10.1109/TAC.1978.1101840.

[31] Bonissone PP, Xue F, Subbu R. Fast meta-models for local fusion of multiple predictive models. Appl Soft Comput 2011;11(2):1529–39. http://dx.doi.org/10.1016/j.asoc.2008.03.006.

[32] Ljung L, Caines PE. Asymptotic normality of prediction error estimators for approximate system models. In: 1978 IEEE conference on decision and control including the 17th symposium on adaptive processes, vol. 17; 1978, p. 927–32. http://dx.doi.org/10.1109/CDC.1978.268066.

[33] Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 1989;257–86.

[34] Rabiner LR, Juang BH. An introduction to hidden Markov models. IEEE ASSP Mag 1986;4–15.

[35] Durbin ESRKA, Mitchison RGJ. Biological sequence analysis: probabilistic models of proteins and nucleic acids. London: Cambridge University Press; 1998.

[36] Genge B, Siaterlis C, Hohenadel M. AMICI: an assessment platform for multi-domain security experimentation on critical infrastructures. In: Critical information infrastructures security. 2013, p. 228–39.

[37] Siaterlis C, Genge B, Hohenadel M. Epic: a testbed for scientifically rigorous cyber-physical security experimentation. IEEE Trans Emerg Top Comput 2013;1(2):319–30. http://dx.doi.org/10.1109/TETC.2013.2287188.

[38] White B, Lepreau J, Stoller L, et al. An integrated experimental environment for distributed systems and networks. ACM SIGOPS Oper Syst Rev 2002;36(SI):255–70.

[39] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst 2011;26(1):12–9.

[40] Cole S, Belmans R. MATDYN, a new matlab-based toolbox for power system dynamic simulation. IEEE Trans Power Syst 2011;26(3):1129–36.

[41] Gungor VC, Sahin D, Kocak T, et al. Smart grid technologies: communication technologies and standards. IEEE Trans Ind Inf 2011;7(4):529–39.

[42] Basseville M, Nikiforov I, et al. Detection of abrupt changes: theory and application, vol. 15. Englewood Cliffs: Prentice Hall; 1993.