Detection of Integrity Attacks in Cyber-Physical Critical Infrastructures Using Ensemble Modeling

Stavros Ntalampiras

***Abstract*—This paper presents an anomaly-based methodology for reliable detection of integrity attacks in cyber-physical critical infrastructures. Such malicious events compromise the smooth operation of the infrastructure while the attacker is able to exploit the respective resources according to his/her purposes. Even though the operator may not understand the attack, since the overall system appears to remain in a steady state, the consequences may be of catastrophic nature with a huge negative impact. Here, we apply a computational intelligent technique which incorporates the merits of two of the heterogeneous modeling approaches (linear time-invariant and neural networks), while considering both temporal and functional dependencies existing among the elements of an infrastructure. The experimental platform includes a power grid simulator of the IEEE 30 bus model and a cyber network emulator. Subsequently, we implemented a wide range of integrity attacks (*replay*, *ramp*, *pulse*, *scaling*, and *random*) with different intensity levels. A thorough evaluation procedure is carried out while the results demonstrate the ability of the proposed method to produce a desired result in terms of false positive rate, false negative rate, and detection delay.**

***Index Terms*—Cyber-physical critical infrastructures (CIs), ensemble modeling, fault diagnosis.**

I. Introduction

NOWADAYS, information and communication technologies (ICTs) play a significant role in monitoring and controlling large-scale infrastructures. Critical infrastructures (CIs) are the assets on which the smooth functioning of a society, economy, etc., depends. Some examples are electricity network including generation, transmission and distribution, gas network for production, transport and distribution, financial services (banking and clearing), transportation systems (fuel supply, railway network, airports, harbors, inland shipping), etc. The usage of an ICT layer offers improved exploitation of the CI and increases the performance of the system while reducing the overall cost (see Fig. 1). The main objectives of such a control structure are as follows: 1) to maintain safe operational goals by limiting the probability of undesirable behavior; 2) to meet the production demands by keeping certain process values within prescribed limits; and 3) to maximize production profit.

Due to security concerns, each CI should be controlled and associated with one ICT network which however increases the cost with respect to installation, maintenance, etc. A relatively easily applicable solution to this issue is the usage of shared infrastructures [1], which nonetheless raises important security concerns. In this context, potential situations of high criticality include cyber threats on the security of Supervisory Control And Data Acquisition (SCADA) systems [2], [3]. A recent paradigm is the Stuxnet worm [4] which was developed for compromising industrial control systems. One should also consider here that the impact of the particular problem raises significantly as the systems tend to have increased demands in terms of occupied space, thus there exist more vulnerable points. Cyber attacks on CIs are rather rare, though they may have a catastrophic social and/or economic impact [5].

The specific problem shares some common characteristics with that of intrusion detection in data packets exchanged among computer systems. In this context, the ultimate aim is to automatically assess data integrity while having only the data content as available information. There are three lines of thought for addressing this problem.

1) Signature-based: Here the algorithm searches for known patterns of malicious activity in the datastream using predefined dictionary of attacks (e.g., [6]).
2) Anomaly-based: This type of approaches estimates characteristic features of the normal behavior and subsequently detects deviations which may appear during an intrusion (e.g., [7]–[9]).
3) Countermeasure-based: The methodologies which belong to this category adapt the involved signals so that the task of intrusion detection is simplified (e.g., [10], [11]).

The first class has limitations since it can only detect *a priori* known attacks while the third one relies on cryptography algorithms where the experience has shown that even the most prominent of them may be crackable [12], [13]. In our opinion, they may be used as a first line of defense as long as the computational complexity is relatively low. The second line of defense should be able to address more complicated cases of malicious events and elaborated attacks. Thus, this work concentrates on the second class and proposes a generic framework for detecting compromised datastreams in cyber-physical CIs.

Here, we briefly describe the literature which is related to this subject. In [14], the researchers present an algorithm for fault detection in power utilities operated via a wide-area broadband network. They use a probabilistic neural network; however, they do not consider the problem of integrity attacks. The authors of [15] design a rule-based scheme using a kernel density estimator and temporal characterization of the network data for detecting integrity attacks. Coutinho *et al.* [7] apply a rough classification algorithm on data coming from a power system control center to derive a set of rules for anomaly detection.

Manuscript received July 22, 2014; revised September 24, 2014; accepted November 02, 2014. Date of publication November 05, 2014; date of current version February 02, 2015. Paper no. TII-14-0767.

The author is with the European Commission, Joint Research Center, Institute for the Protection and Security of the Citizen, 21027 Varese, Italy (e-mail: dalaouzos@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2014.2367322

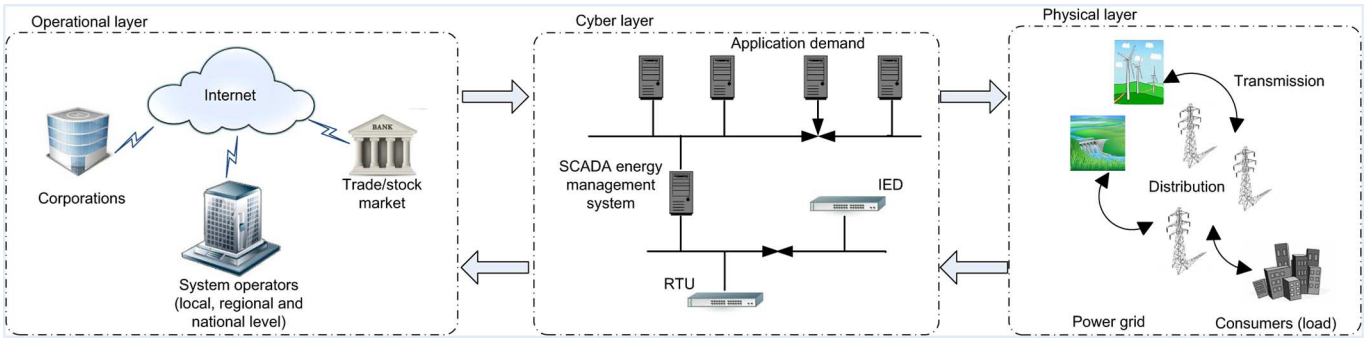

Fig. 1. Interactions of a power control system. The interconnections between the operational, cyber, and physical layer are shown.

However, this algorithm is subject to noise meaning that the derived rules may present great variations in case of interferences, not to mention that it requires a large amount of data. In [16], the authors use an analytical n-gram model for anomaly detection in SCADA systems. They consider simple fault cases without the presence of integrity attacks. Last but not least, Coutinho *et al.* [17] present an intrusion detection system based on the transfer function. This method needs at least one signal to come from a trusted source while it filters alarms and reports rules on the integrity of the data. The authors apply the method on a waste water treatment system.

There does not exist a systematic approach addressing the problem of integrity attack detection as it appears in cyber-physical CIs. Even though the topic has been of interest to the scientific community, most works only address the problem on the cyber layer [18], [19] without considering the interdependent nature of industrial control systems and its increased importance when CIs are at stake. The main property of these systems is that in the SCADA environment, it is unacceptable to have false negatives (FNs) while a low false positive (FP) rate is desired. This paper contains the following novel features.

1) A generic and flexible ensemble modeling algorithm exploiting the merits of linear and nonlinear models.
2) Knowing the underlying analytical relationships of the network under monitoring is not a prerequisite.
3) Exploitation of both temporal and functional relationships exists in the physical layer.
4) Consideration of a wide range of integrity attack patterns.
5) Addressing the problem of biased models by means of a cognitive level.
6) A fault/attack dictionary is not required.

The rest of this article is organized as follows: Section II formulates the problem, while Section III details the proposed integrity attack detection method. Section IV present the experimental framework and provides the analysis of the results. Finally, Section V concludes this work including future research subjects.

II. Problem Formulation

Let us consider a CI composed of N nodes which can be homogeneous, i.e., associated with measurements of same physical quantity (e.g., only voltages), or heterogeneous, i.e., with measurements of different physical quantities (e.g., voltages and currents). We assume that the nodes are indirectly related to each other since they are part of the same entity facilitating a specific service, e.g., a smart-grid network providing electricity to a city. Hence, the acquired datastreams are correlated to some extent. However, we do not assume the existence of any analytical model "explaining" the underlying relationships while the aim of the proposed method is to learn them by mining the incoming data packets. We emphasize that the proposed detection technique does not require any specific network topology or routing protocol.

Let $X_{i,T_0} = \{X_i(t), t = 1, \ldots, T_0\} : \mathbb{N} \to \mathbb{R}$ be the stream of data acquired by the ith node. Without any loss of generality, we assume a scalar-in-time datastream. Let us assume that at an unknown time instant $T^* > T_0$, an abnormal situation (fault and integrity/DoS attack) affects node i. We do not make any assumptions regarding the magnitude or the time profile of the malicious event affecting the process generating the datastream.

The aim of the proposed technique is to detect and identify the occurred situation with the smallest latency and FP and negative rates.

III. Creating the Ensemble of Models by Exploiting Temporal and Functional Redundancies

The first step of the proposed technique is to construct a model capturing the underlying relationships among the datastreams acquired by each node i of the infrastructure. To this end, we rely on a relatively new type of modeling, *ensemble modeling* [20]. Creating an effective ensemble of models is a primary goal of any ensemble technique. Therefore, we exploit the fact that nodes (either homogeneous or heterogeneous) are part of the same infrastructure (or system of infrastructures), and hence, the acquired datastreams are correlated to some extent while the ensemble tries to capture the existing redundancies.

A. Model Hierarchies

In this context, there exist both temporal and functional redundancies among the datastreams. In particular, we intend to exploit the temporal redundancy via modeling the evolution over time of the measurements coming from a specific node, and the functional redundancies existing among different nodes associate either with the same type of variable (homogeneous nodes) or different but correlated one (heterogeneous nodes).

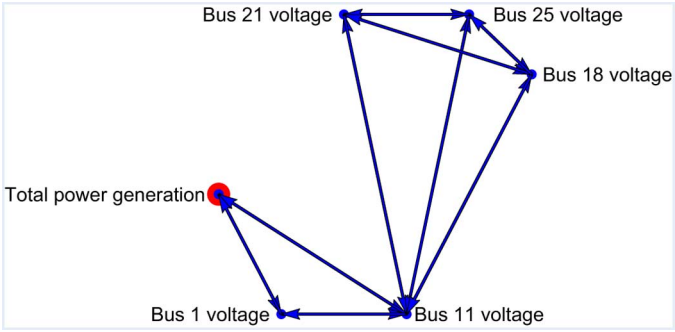

Fig. 2. Reduced dependency graph of the IEEE 30 bus energy model depicting the most relevant relations among the elements of the power network. Only relationships with correlation above a certain threshold may be considered (in this example, threshold = 0.6).

In addition, this work is based on the use of different hierarchies of models to fully explore the prediction abilities of these models. The considered model hierarchies, which are detailed below, encompass both linear input–output models and neural networks. Obviously, this set of model hierarchies can be easily extended to consider, e.g., linear/nonlinear state space models or time-series forecasting models, to name a few.

These two degrees-of-freedom (i.e., temporal/functional redundancy and model hierarchies) provide us with a powerful tool to create an ensemble of models. The mechanism to effectively aggregate these models is detailed in Section III-E.

B. 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. This aims not only at the reduction of the computational cost but also at discarding those relationships between poorly correlated nodes and avoid including "poor" models within the ensemble. As suggested in [21] and [22], poor models could decrease the performance of the overall ensemble. In fact, poorly correlated datastreams reflect a weak functional dependency yielding to poor-performing models which may decrease the overall performance of the ensemble. Last but not least, this way the system is able to operate even when data coming from a specific node are unavailable due to a malfunction, fault, etc. since it does not require data from the entire set of nodes.

The first stage of this preliminary step is the estimation of the cross correlations among the datastreams coming from all N power nodes. Afterward, a dependency graph, which initially includes 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 graph. Fig. 2 shows the reduced dependency graph of the IEEE 30 bus system considered in the Section IV. For visibility reasons, we included data coming from five buses (1, 11, 18, 21, and 25) and the total power generation. In Fig. 2, the arcs connecting CI elements represent highly correlated nodes.

In many real-world applications, a reasonable mathematical model of the system is hard to obtain, thus black-box modeling techniques represent the only feasible solution. For this reason, black-box model hierarchies have been taken into account in this work. Let us now briefly describe the considered model hierarchies.

C. Linear Input–Output Models

These models provide an input–output representation of the process under observation [23]. Specifically, linear input–output models characterize a linear relationship between the considered input and output variables, described in the general MISO canonical form

$$A(z)X_i(k) = \sum_{j=1, j\neq i}^{N} \frac{B_j(z)}{F_j(z)} X_j(k) + \frac{C(z)}{D(z)} d(k) \qquad (1)$$

where z is the time-shift operator, $A(z)$, $B_j(z)$, $C(z)$, $D(z)$, and $F_j(z)$ represent the z-transform functions, X_j is the jth input, and $d(k)$ is an independent and identically distributed (i.i.d.) random variable accounting for the noise.

Among the wide range of linear input–output models, we considered AR and ARMA model for the temporal redundancy and ARX and ARMAX for the functional one.

1) Autoregressive (AR) model

$$A(z)X_i(k) = d(k). \qquad (2)$$

2) Autoregressive moving-average (ARMA) model

$$A(z)X_i(k) = C(z)d(k). \qquad (3)$$

3) Autoregressive with exogenous inputs (ARX) model

$$A(z)X_i(k) = \sum_{j=1, j\neq i}^{N} B_j(z)X_j(k) + d(k). \qquad (4)$$

4) Autoregressive moving-average with exogenous inputs (ARMAX) model

$$A(z)X_i(k) = \sum_{j=1, j\neq i}^{N} B_j(z)X_j(k) + C(z)d(k). \qquad (5)$$

The number of nodes actually considered in the ARX or ARMAX model could be less than N according to the reduced dependency graph.

Let f_1^i, f_2^i, f_3^i, and f_4^i be the estimators of model (2)–(5), see [23], and let $f_1^i(k)$, $f_2^i(k)$, $f_3^i(k)$, and $f_4^i(k)$ be the corresponding predicted values at time k to estimate $X_i(k)$.

D. Neural Networks

Among the wide range of neural network-based solutions for time-series data modeling (e.g., [24], [25]), we consider the reservoir networks (RNs) that represent a novel kind of echo-state networks providing good results in several demanding applications, such as speech recognition [26], saving energy in wireless communication [27], etc. RNs are able to capture nonlinear relationships existing within data coming from various nodes.

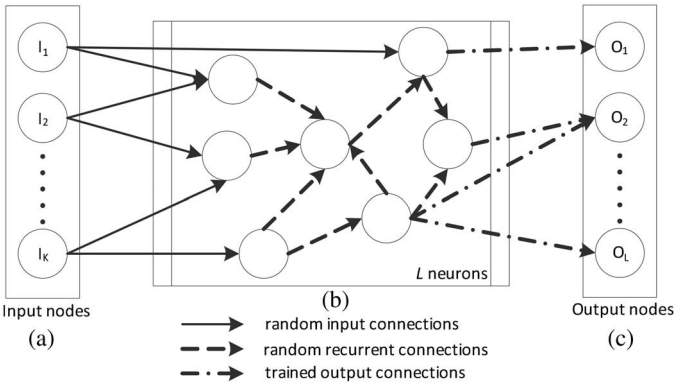

Fig. 3. Standard reservoir network consisting of three layers: (a) the input; (b) the reservoir; and (c) the readout. The second layer includes neurons with nonlinear activation functions. The weights of the input and the recurrent connections are randomly fixed. The weights to the output nodes are the only ones being trained.

A reservoir network, the topology of which is depicted in Fig. 3, includes neurons with nonlinear activation functions which are connected to the inputs (input connections) and to each other (recurrent connections). These two types of connections have randomly generated weights, which are kept fixed both during the training and operational phase. Finally, a linear function is associated with each output node.

Its parameters are the weights of the output connections and are trained to achieve a specific result, e.g., that a given output node produces high values for observations of a particular class. The output weights are learned by means of linear regression and are called read outs since they "read" the reservoir state. More details about the RN training and the echo state property can be found in [28].

As a general formulation of the RNs, we assume that the network has K inputs, L neurons (usually called reservoir size), M outputs (O_L in Fig. 3), while the matrices $W_{\text{in}}(K \times L)$, $W_{\text{res}}(L \times L)$, and $W_{\text{out}}(L \times M)$ include the connection weights. The RN system equations are as follows:

$$x(k) = f_{\text{res}}(W_{\text{in}}u(k-1) + W_{\text{res}}x(k-1)) \tag{6}$$

$$y(k) = f_{\text{out}}(W_{\text{out}})x(k) \tag{7}$$

where $u(k)$, $x(k)$, and $y(k)$ denote the values of the inputs, reservoir outputs, and the read-out nodes at time k, respectively. f_{res} and f_{out} are the activation functions of the reservoir and the output nodes, respectively. In this work, we consider $f_{\text{res}}(x) = \tanh(x)$ and $f_{\text{out}}(x) = x$, and we fix $M = 1$ since we are considering single-output models.

Linear regression is used to determine the weights W_{out}

$$W_{\text{out}} = \underset{W}{\text{argmin}} \left(\frac{1}{N_{tr}} \|\boldsymbol{XW} - \boldsymbol{D}\|^2 + \epsilon \|W\|^2 \right) \tag{8}$$

$$W_{\text{out}} = (X^T X + \epsilon I)^{-1}(X^T \boldsymbol{D}) \tag{9}$$

where $\boldsymbol{XW}$ and $\boldsymbol{D}$ are the computed vectors, I is a unity matrix, N_{tr} is the number of the training samples, and ϵ is a regularization term.

The recurrent weights are randomly generated by a zero-mean Gaussian distribution with variance v, which essentially controls the spectral radius (SR) of the reservoir. The largest absolute eigenvalue of W_{res} is proportional to v and is particularly important for the dynamical behavior of the reservoir [29], [30]. W_{in} is randomly drawn from a uniform distribution $[-\text{InputScalingFactor}, +\text{InputScalingFactor}]$, which emphasizes/deemphasizes the inputs in the activation of the reservoir neurons. It is interesting to note that the significance of the specific parameter is decreased as the reservoir size increases.

To model the temporal redundancy, (6) can be reformulated by substituting $y(k)$ with $X_i(k)$ and $u(k-1)$ with $[X_i(k-1), \ldots, X_i(k-n_i)]$. Similarly, to model the functional redundancy, $y(k)$ must be substituted with $X_i(k)$ but $u(k-1)$ collects data coming from other nodes of the network excluding the ith sensor, i.e., $[X_1(k), \ldots, X_i(k-n_1), \ldots, X_N(k), \ldots, X_N(k-n_N)]$. Even in this case, the dependency graph could reduce the number of nodes to be actually considered as inputs on the RN.

Similar to what presented for the linear input–output models, we define f_5^i and f_6^i as the RN predictors for the temporal and functional redundancies, respectively, and let $f_5^i(k)$ and $f_6^i(k)$ be the corresponding predicted values for the ith sensor modeling at time k.

E. Aggregating Models Within the Ensemble

While Section II-D aimed at describing how to create a set of models for data reconstruction, we here exploit the concept of "ensemble of models" $E = \{f_1^i, f_2^i, \ldots, f_M^i\}$, where several model estimators are combined to obtain a better estimation than each model could achieve alone. As described in [21] and [22], an effective aggregation mechanism plays a crucial role in the generalization ability of the ensemble estimation.

Aggregating by means of a weighted average of the estimators is common in [21] and [22, Ch. 4], i.e.,

$$\hat{X}_i(k) = \bar{f}^i(k) = \sum_{m=1}^{M} \omega_m f_m^i(k) \tag{10}$$

where M is the number of models within the ensemble (in our case $M = 6$) and ω_m's represents the weights associated with the estimators. These are generally constrained on

$$0 \leq \omega_m \leq 1, \quad m = 1, \ldots, M \tag{11}$$

$$\sum_{m=1}^{M} \omega_m = 1. \tag{12}$$

Obviously, the choice of the weights is critical for the ensemble performance. Several approaches can be found in the literature ranging from the simple average (where $\omega_m = 1/M$) to the Akaike's weights [31]. By exploiting the Lagrange multiplier method, Perrone and Cooper [21] provide the closed-form solution to the optimal weights for an ensemble of predictors in case of regression problems. Unfortunately, this solution requires the correlation matrix of the estimators to be invertible, an hypothesis difficult to satisfy in ensembles of estimators which are generally highly correlated (yielding to singular

or ill-conditioned correlation matrix). Interestingly, prediction selection [e.g., see [22, Ch. 6]] can be considered a subcase of weighted average where the weight of the discarded predictors is 0, and the remaining predictors share the same weight.

A straightforward solution to the weight identification problem is represented by the numerical optimization of the ensemble reconstruction error, i.e., $X_i(k) - \bar{f}^i(k)$, on a validation set V_S

$$\left[\bar{\omega}_1^i, \ldots, \bar{\omega}_M^i\right] = \underset{\omega_1,\ldots,\omega_M}{\operatorname{argmin}} \frac{1}{|V_S|} \sum_{k=1}^{|V_S|} \left(X_i(k) - \bar{f}^i(k)\right)^2 \quad (13)$$

where the weights $\left[\bar{\omega}_1^i, \ldots, \bar{\omega}_M^i\right]$ are identified as those minimizing the estimation error of $\bar{f}^i(k)$ on a validation set V_S.

Next, we suggest two approaches for the identification of the optimal weights differentiating in both the aggregation mechanism and the learning strategy. The *first approach* is based on a joint selection/weighting approach that explicitly exploits temporal and functional redundancies. To this purpose, the best temporal model and the best functional model are identified as those providing the lowest reconstruction error on the validation set V_S. In more detail, in case of temporal redundancy, we compute

$$\hat{e}_{t^*}^{i,V_s} = \sum_{k=1}^{|V_S|} \left(X_i(k) - f_{t^*}^i(k)\right)^2 \quad (14)$$

and let $t^* \in \{1, 2, 5\}$ be the temporal estimator providing the lowest reconstruction error. Similarly, for the functional redundancy, we compute

$$\hat{e}_{f^*}^{i,V_s} = \sum_{k=1}^{|V_S|} \left(X_i(k) - f_{f^*}^i(k)\right)^2 \quad (15)$$

and let $f^* \in \{3, 4, 6\}$ be the functional estimator providing the lowest reconstruction error. We prune the ensemble $E = \{f_1^i, f_2^i, \ldots, f_M^i\}$ keeping only the best temporal and functional models, i.e., $E^P = \{f_{t*}^i, f_{s*}^i\}$. We then optimize the weights of the pruned ensemble $\bar{f}_P^i(k) = \omega_{t*}^i f_{t*}^i(k) + \omega_{s*}^i f_{s*}^i$ as follows:

$$\left[\bar{\omega}_{t*}^i, \bar{\omega}_{s*}^i\right] = \underset{\omega_{t*}^i, \omega_{s*}^i}{\operatorname{argmin}} \frac{1}{|V_S|} \sum_{k=1}^{|V_S|} \left(X_i(k) - \bar{f}_P^i(k)\right)^2. \quad (16)$$

The *second approach* is based on the ability to capture nonlinear relationships among the estimators' outputs within the ensemble. Different from the previous approach, the ensemble is not pruned and the aggregation mechanism is more complex than the weighted average of the outputs of the predictors [i.e., here the ensemble does not follow (10)]. To this purpose, we define a function Φ as follows:

$$\hat{X}_i(k) = \bar{f}_\Phi^i(k) = \Phi\left(f_1^i(k), f_2^i(k), \ldots, f_M^i(k)\right). \quad (17)$$

In principle, Φ can be any nonlinear possibly time-dependent function of the predictors outputs. In this work, function Φ has been implemented through a RN whose input at time k is $\{f_1^i(k), f_2^i(k), \ldots, f_M^i(k)\}$ and whose output $\bar{f}_\Phi^i(k)$ is the ensemble estimate of the missing datum $X_i(k)$. The weights of the RN are trained on the validation set V_S. It should be noted that the modularity of the proposed estimator allows the consideration of other implementations of Φ as well, e.g., feed-forward neural networks, recurrent neural networks, or nonlinear regression functions.

Algorithm 1. An abnormality detection framework based on ensemble modeling

1. Build the ensemble $\mathcal{M}$ on an input training data;
2. Initialize the estimation E;
repeat
   3. Acquire input data $X(t)$;
   4. Compute the estimation $E(t)$;
   5. **if** $|X(t) - E(t)| < T_a$ **Then**
      An abnormal situation is detected.
   end
   6. $t = t + 1$;
until *(1)*;

F. Detection of Abnormalities via the Ensemble Model

The ensemble of models in the form of either linear combination of the best temporal and functional models or the nonlinear full combination is used for detection of abnormalities in an identical two-step process (also shown in Algorithm 1): 1) compute the estimation $E(t) \in \{\bar{f}_\Phi^i(t), \bar{f}_P^i(t)\}$; and 2) compare it with the threshold T_a which is calculated using the validation dataset V_s. In case, the observed discrepancy ($|X(t) - E(t)|$) is greater than T_a the specific datum is associated with an abnormality. In the opposite case, the datum is considered to belong to the nominal operating state and the associated data are used to adapt T_a so that its value is updated and avoid misinterpreting the normal data for abnormal one.

G. Cognitive Level

Using the methodology described above the system makes a decision on the validity of data coming from each node (bus). A detection of an abnormal event at this level can be associated either with a true abnormal event of with a model bias. At the latter case, a cognitive level is required for gathering decisions concerning individual buses and assessing a possible model bias. The rationale of the cognitive level is that an integrity attack on a specific bus will affect the rest of the buses, even momentarily, since the network will have to adjust in order to continue operating normally, e.g., reallocate the load for satisfying the demand of the consumers. At this time instance, other datastreams or at least the ones highly correlated to the one suffering the attack will present abnormalities. Thus the algorithm running at the cognitive level examines the existence of potential abnormalities existing in the rest of the network. If this hypothesis is true then the abnormal datum is associated with an attack. On the opposite case, the specific ensemble is considered biased and it is retrained online using the incoming data. The process is shown in Algorithm 1.

IV. Experimental Test-Bed and Results

The experimental test-bed is based on the IEEE 30-bus model [32] (see Fig. 4) where detection and identification of infrastructure states deviating from the nominal one is not a trivial task due to the complexity of the network. AMICI framework [33], which encompasses both MatPower [34] and MatDyn [35], was used for the simulation of the model. Ref. [36] was employed for emulating the distributed control system (SCADA servers and corporate network) shown in Fig. 1. It should be mentioned that during the nominal state, the load fluctuates between margins predefined by the IEEE model while the sampling period is 20 ms.

The power grid is operated via the ICT layer which includes algorithms for transferring data to the central controller. Subsequently, the control logic code is executed by the PLCs which are the elements affected by integrity attacks.

A. Compromising the Integrity of the Infrastructure

In this work, we assume that the attacker has full access to the infrastructure meaning that he has the ability to hijack, record, and replay/alter node data according to his best interest while compromising the operation of the overall framework (man-in-the-middle attack). We further assume that the attacker may monitor the infrastructure network for a long period of time, thus he is familiar with its dynamic behavior as well.

In essence, the attacker is able to alter the control inputs of the smart grid. As a result, the controller does not get reliable information and the system becomes an open loop. The only way to deal with such cases is to promptly detect the attack.

Based on the above described logic and the kinds of attacks considered in the literature (see Section I) in this paper, we encompass a generic set of integrity attacks representing a wide range of scenarios.

1) *Pulse*: In this case, the datastream is altered according to an additive pulse: $X_i^*(t) = X_i(t) + \mathrm{rect}(t)$, where $X_i^*(t)$ is the compromised data, $X_i(t)$ the data coming from the nominal network state as recorded by the attacker, and

$$\mathrm{rect}(t) = \begin{cases} 0, & \text{if } |t| > \frac{1}{2} \\ a_p/2, & \text{if } |t| = \frac{1}{2} \\ a_p, & \text{if } |t| < \frac{1}{2} \end{cases}$$

where a_p is the attack parameter.

2) *Scaling*: Here the recorded measurements are scaled on the basis of the parameter a_p: $X_i^*(t) = a_p \times X_i(t)$.
3) *Ramp*: During this attack type the recorded measurements are gradually modified by adding a ramp function with parameter a_r: $X_i^*(t) = X_i(t) + \mathrm{ramp}(t)$, where $\mathrm{ramp}(t) = a_r \times t$.
4) *Random*: This type of attack suggests summing the recorded datastream with a uniform random distribution from the interval (a, b): $X_i^*(t) + \mathrm{rand}(a, b)$.
5) *Replay*: The final type of attack which is usually found in the literature with the descriptive name *replay* merely involves the identical repetition of *a priori* recorded data.

Algorithm 2. The algorithm running at the cognitive level for differentiating between an integrity attack and a model bias.

1. Collect the decisions made for each bus
$\mathcal{D}(t) = \{D_1(t), \ldots, D_{30}(t)\}$, where

$$D_i = \begin{cases} 0 & \text{no abnormality detected on bus } i \\ 1 & \text{abnormality detrected on bus } i \end{cases}$$

repeat
  2. **for** *i=1:30* **do**
    3. **if** $D_i(t) == 1$ **then**
      4. **If** $D_{corr}(t) == 1$ **then**
        Alert: Integrity attack on Bus i
      else
        Ensemble i is biased
      end
    end
  end
  5. $t = t + 1$;
until *(1);*

It should be emphasized that the attacker may compromise the network using the same or a different attack type, i.e., he may start with a specific kind of attack and continue with another one.

B. Parameterization of the Experimental Framework

The order of each linear time-invariant model (AR, ARMA and ARMAX) was selected as the one providing the best reconstruction results on the validation set V_S while the orders belonged to the {3,4,5,6} set. With respect to the RN its parameters were selected by means of exhaustive search. They were taken from the following sets: $\mathrm{SR} \in \{0.8, 0.9, 0.95, 0.99\}$; $L \in \{100, 500, 1000, 5000, 10000\}$; and $\mathrm{InputScalingFactor} \in \{0.1, 0.5, 0.7, 0.95, 0.99\}$. The implementation of the RN was based on the echo state network toolbox which is available at http://reservoir-computing.org/software (date last viewed 30-06-2014).

Before selecting the parameters of the integrity attack types (a_p, a_s, a_r) one should consider that they play a very critical role meaning that the data are continuously passing through data quality control checks. Thus the attacker must be extremely careful not to trigger any kind of alarm mechanism. The parameters should be chosen so that the attack has an impact on the system and at the same time overall network remains stable. After experimenting with these conditions on the framework presented in Fig. 4, we decided to use the following parameter to realize the integrity attacks: $a_s \in \{0.02, 0.03, 0.04, 0.05\}$; $a_r \in \{0.02, 0.03\}$; $a_p \in \{0.02, 0.04\}$ and random $a = 0.3, b = 0$.

Each scenario had a duration of 35 000 samples while the attack was injected at sample 10 000. The first 4000 samples were used for training the model while the following 4000 for validation. The rest were used for testing while the results are averaged over all types of integrity attacks while each one with a specific parameter was executed 50 times.

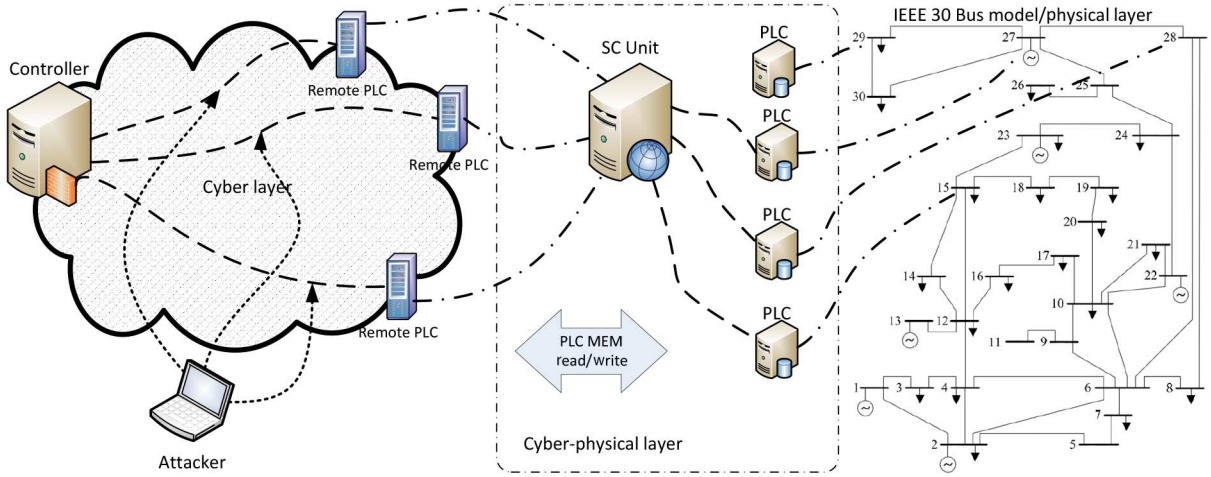

Fig. 4. Experimental architecture showing the cyber, cyber-physical, and physical layer.

TABLE I
DETECTION RESULTS OF THE PROPOSED METHOD

Model type	FP (%)	FN (%)	Delay (# of samples)
Temporal modeling	5.2	3.8	34
Functional modeling	6.3	6.9	36
Linear Combination of Temporal and Functional modeling	3.2	3.5	32
Nonlinear Combination of Temporal and Functional modeling	1.5	2.9	21

We employed three figures of merits for evaluating the detection accuracies.

1) FP counts the times the algorithm detects an attack in the datastream when there is not one.
2) FN counts the times the algorithm does not detect an attack while there is one.
3) Detection delay (DD) measures the time delay (in samples) that is needed by the algorithm to detect an attack.

C. Experimental Results

We compared several methodologies on the same experimental platform. In Table I, we tabulate the results according to each modeling type: 1) temporal; 2) functional; 3) linear combination of the temporal and functional models [based on (13)]; and 4) nonlinear combination of the temporal and functional models [based on (17)].

We observe that modeling the data based on temporal redundancies alone provides better figures of merit with respect to the functional modeling. This fact is expected due to the characteristically periodic character of the datastreams coming from a smart grid where user demand and load variation follow consistent temporal patterns. However, the functional redundancies are also useful since they capture relationships existing within different buses which may be strong, i.e., associated with high cross correlation values (see Section III-B). Interestingly when combined, these two types of modeling provide better performance with respect to all figures of merit, i.e., FP (3.7% decrease), FN (1.1% decrease), and detection delay (13 samples decrease). This performance increase demonstrates that these modeling types capture redundancies with complementary characteristics. Moreover, their nonlinear combination reaches the lowest metrics achieving quite encouraging results given the complexity of the task.

Overall, we infer that an ensemble of models is able to better capture the dependencies existing in the dataset than any single modeling method. However, it is important to note that we avoid using "poor" models in the ensemble by means of a correlation map. Discarding models with low reconstruction ability ensures higher ensemble performance. We conclude that ensemble modeling can effectively address the problem of integrity attack detection in cyber-physical CIs. In addition, the proposed method is able to be applied to system.

V. CONCLUSION AND FUTURE DIRECTIONS

This paper addresses a quite challenging task, that of detecting compromised data belonging to an ICT-controlled CI. To this end, we employed a pool of linear and nonlinear models which was thoroughly evaluated. The ensemble exploits temporal and functional dependencies as well as two aggregation algorithms while the nonlinear fusion provided the lowest figures of merit, i.e., the best performance. Overall the superiority of the ensemble approach was evident. The method can be applied to simpler cases of attacks, such as denial of service or faults. Furthermore, the proposed method is flexible enough to handle different types of (critical) infrastructures. An interesting point is that heterogeneous types of models fitting better the problem specifications can be easily incorporated. In addition, the proposed ensemble modeling method can be applied unaltered to energy models with different levels of complexity.

In the future, we wish to explore the following directions.

1) Design a technique for data *forecasting* which will be activated after the attack detection and serve the controller so that its operation may continue smoothly.
2) Implement a statistical similarity measure (based on Kullback divergence) in order to group different types of detected attacks. The methodology is capable of detecting unknown types of attacks as long as they force the

system to exhibit a statistically different behavior which is a reasonable assumption. In case the attacker is using statistically similar patterns, the system will be able to group the data associated with these attacks, model them and identify them in the future. This is an important aspect leading to the creation of an attack dictionary which may help the investigation of a potential catastrophic situation.

3) Finally, we intent to cooperate with a power grid operator to apply the proposed method and thoroughly study its limitations (e.g., dealing with wormhole type of attacks which now may remain undetected).

REFERENCES

[1] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 5–20, Feb. 2013.

[2] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for SCADA systems," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1836–1846, Nov. 2008.

[3] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Trans. Syst., Man Cybern. A*, vol. 40, no. 4, pp. 853–865, Jul. 2010.

[4] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May 2011.

[5] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber-physical system security for the electric power grid," *Proc. IEEE*, vol. 100, no. 1, pp. 210–224, Jan. 2012.

[6] H. Zhengbing, L. Zhitang, and W. Junqi, "A novel network intrusion detection system (nids) based on signatures search of data mining," in *Proc. 1st Int. Workshop Knowl. Discov. Data Mining (WKDD'08)*, Jan. 2008, pp. 10–16.

[7] M. Coutinho *et al.*, "Anomaly detection in power system control center critical infrastructures using rough classification algorithm," in *Proc. 3rd IEEE Int. Conf. Digital Ecosyst. Technol. (DEST'09)*, Jun. 2009, pp. 733–738.

[8] X. He, Z. Wang, Y. Liu, and D. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1670–1679, Aug. 2013.

[9] J. Neuzil, O. Kreibich, and R. Smid, "A distributed fault detection system based on IWSN for machine condition monitoring," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1118–1123, May 2014.

[10] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on scada systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1396–1407, Jul. 2014.

[11] A. Giani *et al.*, "Smart grid data integrity attacks," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1244–1253, Sep. 2013.

[12] X. Wang, D. Feng, and X. Yu, "An attack on hash function Haval-128," *Sci. China F. Inf. Sci.*, vol. 48, no. 5, pp. 545–556, 2005.

[13] X. Wang, Y. Yin, and H. Yu, "Finding collisions in the full sha-1," in *Proc. Adv. Cryptol. (CRYPTO'05)*, vol. 3621. Berlin, Germany: Springer-Verlag, 2005, pp. 17–36.

[14] S. Su, X. Duan, X. Zeng, W. L. Chan, and K. K. Li, "Context information based cyber security defense of protection system," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Jun. 2007, p. 1.

[15] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 580–591, Mar. 2014.

[16] J. Bigham, D. Gamez, and N. Lu, "Safeguarding scada systems with anomaly detection," in *Proc. MMM-ACNS*, vol. 2776. New York, NY, USA: Springer, 2003, pp. 171–182.

[17] S. Papa, W. Casper, and S. Nair, "A transfer function based intrusion detection system for SCADA systems," in *Proc. IEEE Conf. Technol. Homeland Sec. (HST)*, Nov. 2012, pp. 93–98.

[18] S. Singh and S. Silakari, "An ensemble approach for cyber attack detection system: A generic framework," in *Proc. 14th ACIS Int. Conf. Softw. Eng. Artif. Intell., Netw. Parallel/Distrib. Comput.*, Jul. 2013, pp. 79–84.

[19] N. Ye, Y. Zhang, and C. Borror, "Robustness of the Markov-chain model for cyber-attack detection," *IEEE Trans. Rel.*, vol. 53, no. 1, pp. 116–123, Mar. 2004.

[20] C. Alippi, "Learning in nonstationary and evolving environments," in *Intelligence for Embedded Systems*. New York, NY, USA: Springer, 2014, pp. 211–247.

[21] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," in *ANNs for Speech and Vision*, R. J. Mammone, Ed. London, U.K.: Chapman & Hall, 1993.

[22] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: Chapman & Hall, 2012.

[23] L. Ljung, Ed., *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1999.

[24] S. Chiewchanwattana, C. Lursinsap, and C.-H. Chu, "Time-series data prediction based on reconstruction of missing samples and selective ensembling of FIR neural networks," in *Proc. Neural Inf. Process. (ICONIP'02)*, Nov. 2002, vol. 5, pp. 2152–2156.

[25] P. Panagi and M. Polycarpou, "A coordinated communication scheme for distributed fault tolerant control," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 386–393, Feb. 2013.

[26] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Reservoir-based techniques for speech recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN'06)*, Jul. 2006, pp. 1050–1053.

[27] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[28] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.

[29] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach," Fraunhofer Institute AIS, Sankt Augustin, Germany, Tech. Rep. 159, 2002.

[30] D. Verstraeten, B. Schrauwen, M. DHaene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Netw.*, vol. 20, no. 3, pp. 391–403, 2007.

[31] Z. Zhao, Y. Zhang, and H. Liao, "Design of ensemble neural network using the akaike information criterion," *Eng. Appl. Artif. Intell.*, vol. 21, no. 8, pp. 1182–1188, 2008.

[32] J. Wen, L. Jiang, Q. Wu, and S. Cheng, "Power system load modeling by learning based on system measurements," *IEEE Trans. Power Del.*, vol. 18, no. 2, pp. 364–371, Apr. 2003.

[33] B. Genge, C. Siaterlis, and M. Hohenadel, "Amici: An assessment platform for multi-domain security experimentation on critical infrastructures," in *Proc. 7th Int. Conf. Critical Inf. Infrastruct. Sec.*, 2013, vol. 7722, pp. 228–239.

[34] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[35] S. Cole and R. Belmans, "Matdyn: A new MATLAB-based toolbox for power system dynamic simulation," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1129–1136, Aug. 2011.

[36] B. White, "An integrated experimental environment for distributed systems and networks," in *Proc. Oper. Syst. Des. Implement. 2002 (OSDI'02)*, Boston, MA, USA, Dec. 2002, pp. 255–270.

Stavros Ntalampiras received the Engineering and Ph.D. degrees in electrical and computer engineering from the Department of Electrical and Computer Engineering, University of Patras, Patras, Greece, in 2006 and 2010, respectively.

Later, he joined the Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Milano, Italy. Since 2013, he has been conducting research at the Joint Research Center of European Commission, Varese, Italy. He has authored over 40 publications in peer-reviewed journals and conferences with at least 200 citations. His research interests include content-based signal processing, fault diagnosis, audio pattern recognition, and computer audition.