



Article

Evidence-Based Analysis of Cyber Attacks to Security Monitored Distributed Energy Resources

Davide Cerotti ¹, Daniele Codetta-Raiteri ¹, Giovanna Dondossola ², Lavinia Egidi ¹,
Giuliana Franceschinis ¹, Luigi Portinale ¹ and Roberta Terruggia ^{2,*}

¹ DiSIT, University of Piemonte Orientale, 15121 Alessandria, Italy; davide.cerotti@uniupo.it (D.C.); daniele.codetta@uniupo.it (D.C.-R.); lavinia.egidi@uniupo.it (L.E.); giuliana.franceschinis@uniupo.it (G.F.); luigi.portinale@uniupo.it (L.P.)

² Transmission and Distribution Technologies Department, RSE Ricerca Sistema Energetico, 20134 Milano, Italy; giovanna.dondossola@rse-web.it

* Correspondence: roberta.terruggia@rse-web.it

Received: 15 June 2020; Accepted: 2 July 2020; Published: 9 July 2020



Abstract: This work proposes an approach based on dynamic Bayesian networks to support the cybersecurity analysis of network-based controllers in distributed energy plants. We built a system model that exploits real world context information from both information and operational technology environments in the energy infrastructure, and we use it to demonstrate the value of security evidence for time-driven predictive and diagnostic analyses. The innovative contribution of this work is in the methodology capability of capturing the causal and temporal dependencies involved in the assessment of security threats, and in the introduction of security analytics supporting the configuration of anomaly detection platforms for digital energy infrastructures.

Keywords: distributed energy resources; cyber threats; early evidence-based anomaly detection; time-driven attack analysis; countermeasures; security analytic; security monitoring; attack forecasting; dynamic Bayesian networks; MITRE ATT&CK

1. Introduction

In recent years, energy infrastructure has been evolving from a traditional architecture where the communications are intra-operator and private, to a new landscape that requires advanced functionalities such as distributed energy resource (DER) control, demand response from flexible loads, and electric vehicle charging management. The evolution is supported by new information and communication technology (ICT) solutions involving end devices, algorithms, and communication networks through heterogeneous and possibly third party infrastructures and services. In the same network, new components have to interact with legacy ones with no or limited cyber security measures. In this setting, the exploitable attack surface widens and the cyber security management becomes a core process.

In European member states, essential service operators (including energy operators) are required to implement the European Network and Information Security (NIS) Directive 2016/1148 [1] whose compliance implies developing methodologies to prevent cyber attack processes to succeed by detecting ongoing attack steps and implementing fast response and defence measures. It is of paramount importance that energy organisations and their employees, within the corporate offices, information technology (IT) and operational technology (OT) departments, and field engineers, are aware of possible threats that may target their cyber-power infrastructures and are prepared to cope with the expected evolution of cyber attacks.

This paper presents a methodology based on artificial intelligence techniques, in particular, dynamic Bayesian networks (DBNs) [2], modelling the cause–effect connections between possible attack techniques deployed by adversaries and observable analytics. Factually relating observable events, such as system and application logs, to unknown adversarial activity, enable anticipating attack behaviours for a prompt and effective defence. A thorough understanding of such relations is necessary to design a cost-effective infrastructure for evidence collection that can help facing adversarial threats.

Detection of adversarial activity is a tricky task due to the multifaceted actions that an attacker can take and in relation to complexities of cyber-physical control systems and their maintenance. A simplistic example is the detection of a malicious alteration of the control software: it must be carefully distinguished from a regular software upgrade in terms of generated alerts. Such issues bring forward the necessity of identifying correctly which aspects must be monitored (and how) because they are more revealing than others, and how to combine the information gathered from different sensors so as to draw a meaningful picture of what is going on in the system. In this spirit, our work aims first of all at exploiting the existing knowledge on attacks in a model capable of rendering in a realistic fashion the possible attack scenarios, in order to understand what the relevant evidence is that must be collected from DERs and the central SCADA controller.

The next step will then be validation in a test laboratory of the modelling results; for this step we implement a framework for attack emulation and detection, integrated in a grid telecontrol setup connecting DER control networks. We plan to learn from this second phase whether our choices are adequate and thus get a feedback to improve the model. When our design choices for the monitoring system will be satisfactory and stable, then our proof-of-concept will be ready for the validation on field data, prior to its deployment in a real operational environment.

One of the contributions of the proposed approach is the flexibility of our modelling methodology: it can be used either in a predictive direction, to forecast from evidence of attackers' activities the most likely evolution of the adversarial process, or in a diagnostic direction, providing support for evaluating the system security level and for identifying the weakest points so that the most appropriate countermeasures can be implemented.

A second key trait of our approach is that it is able to cover the whole architecture of DER control across several security domains, capturing each domain's peculiarities and considering attack processes spreading from a corporate IT network down to a process control OT one. Our reference attack processes involve two crucial adversarial phases: the compromise of a device on the operational network from an initial foothold in the corporate one, followed by an attack phase aimed at destabilising the electrical process. In our previous work [3] we analysed the different phases of such attack processes. In this work the analysis focuses on the last attack phase, going into the time dimension of the attack steps, a very relevant feature for designing anomaly response strategies that meet the response times of DER control loops. We illustrate the methodology with a proof-of-concept attack setting, analysing a restricted range of possible techniques that can be deployed by the attacker for disturbing the DER control loop. We ground our analysis on the MITRE ATT&CK knowledge base of real world cyber incidents and the MITRE Cyber Analytics Repository (CAR) [4]. We enrich and specialise the techniques and analytics for the energy setting by also deriving data from ICS-CERT advisories [5] using the common vulnerability scoring system (CVSS) [6] scores.

We demonstrate the flexibility of our methodology for both diagnostic and predictive analyses.

1.1. DER Control and Attack Processes

A digital energy infrastructure is composed of several ICT areas with different functionalities and goals, structured in a layered architecture. Components within each area communicate with each other by means of local networks, but also with remote devices in other zones exploiting possibly heterogeneous technologies. The inter-area connections introduce additional vulnerabilities while extending the impact of existing ones over multiple domains with diverse resilience strategies

and security policies. A key principle to secure digital energy architectures is segmenting the various environments into different trust levels.

This paper addresses the analysis of cyber attack processes targeting an OT architecture of significant grid users (SGU) with flexible energy resources possibly controlled by transmission system operators (TSO), distribution system operators (DSO), or aggregators. The DER domain is particularly relevant in the energy transition scenarios targeting the long term objectives till 2050, as stated at the European level by the clean energy package, and by each national energy and climate integrated plan. In terms of security, this domain extends the IT/OT/ICS perimeter of a given grid operator with further network branches owned by DER operators or aggregators with lower security capacity. The specifications of the suitable cyber security requirements and measures, as security monitoring systems, for protecting the extended perimeter by attacks to DER networks, are the subject of future energy sector regulations.

Attack processes are not atomic actions but can be decomposed in several attack phases (see, e.g., [7,8]) and their realisation can last even several years. Real attacks on energy systems, such as Stuxnet [9], and more recently the attacks on the Ukrainian power grids (BlackEnergy 2 and industroyer/CrashOverride) [10,11], start with the compromise of a node in the corporate network, and then the attacker moves from the enterprise area to the OT network to reach the final target. In the process, the attacker leaves traces that can be used to intercept malicious activities in the initial phases, before the attacker reaches the most critical assets. The knowledge of the attack structure is of great help for the implementation of defence strategies.

As we detail in the next sections, our analysis centres on attack processes targeting SGU OT networks (Figure 1) which are typically characterised by a plant controller with external and internal communication interfaces. The external interfaces allow the operators to monitor and control the plant behaviour, and also to remotely perform maintenance activities. Through the internal interfaces the plant controller reaches the field devices, e.g., smart meters and inverters, in charge of getting the electrical measures and actuating setpoints. This paper analyses the last phase of attack processes on SGU OT networks starting from a compromised plant controller, and by including in the model security knowledge from the DER domain, it shows how the DBN model supports the attack forecast and diagnosis by correlating the attack techniques with observable events.

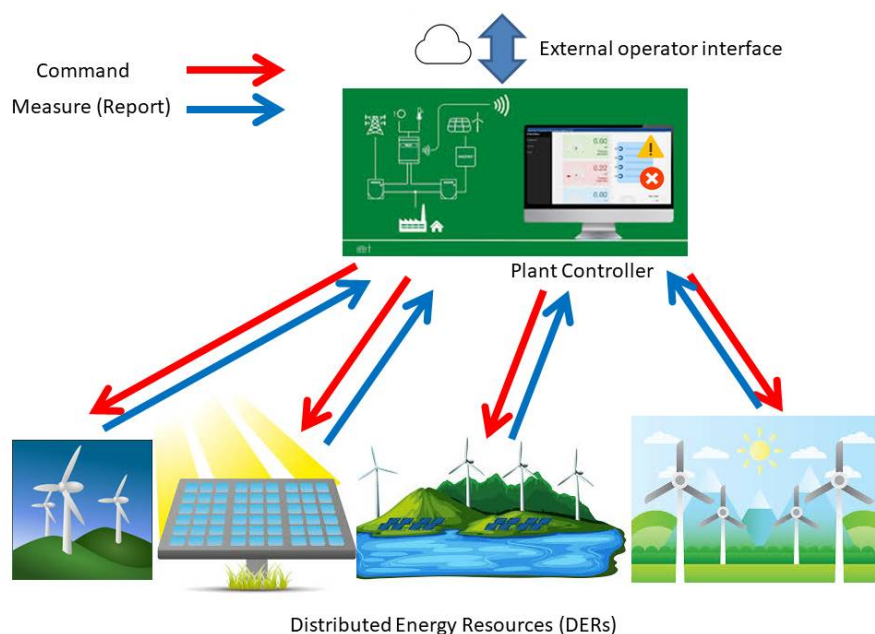


Figure 1. Reference architecture.

1.2. Related Work

A survey on the main technologies used in power systems, the major vulnerabilities, and the cybersecurity requirements, is presented in [12]. The impact of cyber attacks to DER connecting grids has been assessed in [13]. Several works present analysis methodologies and strategies to mitigate the cyber risks in critical systems. In [14] the methodology models the main IT components, attack steps, and countermeasures to identify the critical assets and secure the supervisory control and data acquisition (SCADA) system. In [15] ensemble classifiers learn the (low entropy) benign traffic in a smart grid to later detect anomalies. In [16] an attack chain designed for industrial Internet of Things (IoT) environments is introduced: it considers the multilevel architecture of an ICS, and uses machine learning classification techniques to map security alerts to the attacks phases. In [17] graphs are used to model cybersecurity properties on the smart grid and to evaluate the effectiveness of security mechanisms. Detection systems in the power grid are mostly rule and pattern-matching recognition oriented [18] and protocol whitelisting [19]. On the other hand, there is a very recent line of research [20] dynamically collecting the SCADA information from phasor measurement units through the wide area monitoring system, for real-time, high granularity monitoring to increase the stability of the power system. This information could be useful to the security analyst to integrate existing detection systems with application level intelligence.

Probabilistic models have been adopted for security level assessment. For example, in [21] several attack objectives have been analysed in terms of vulnerabilities, modelling the SCADA weaknesses by means of attack trees (AT); and in [22] an AT formulation representing power system control networks is used to evaluate system vulnerabilities and attack goals. Using attack graphs (AG), the model of the attack must not follow a tree structure and can include the characteristics of the adversary [23].

In this paper the preliminary model of the attack scenario is an AG; the use of ATs instead of AGs could be justified by the presence of only one final goal in the scenario, but sequential order constraints on events cannot be represented in standard ATs, where in contrast, events are independent and can be combined only using Boolean gates (AND/OR).

Bayesian networks (BN) [2] and their variants are inherently interpretable models, meaning that it is possible to ascertain why a conclusion was reached and validate how model changes affect the inference results, in contrast with "black box" models such as neural networks [24] where such a possibility is typically precluded and is an open problem in the explainable AI [25].

The adoption of BNs has been advocated by several works in the field of security assessment for critical infrastructures; for example, in [26] a BN model of the system under attack is derived from the AT of the same system; in our paper we resort to AG instead of AT for the reasons expressed above. In [27] the initial parameters of the BN are elicited by means of CVSS and predefined discrete probability levels (certain, probable, expected, unlikely, improbable, impossible). In [28] an algorithm for BN inference (analysis) is adapted to be applied to an AG; this permits avoiding the constraint of no cycles in the BN graph, which can be present in the AG. In [29] the authors propose a cyber-to-physical risk assessment model to quantify the impact of cyber threats on physical process safety in ICS. Said model was based on the analysis of BNs where the initial parameters are derived from CVSS scores.

In all these works, differently from our approach, the model is a standard (static) BN, and thus not able to cope with the temporal evolution of the system, and the main objective is risk assessment; hence, it does not exploit the capability of BNs to support an analyst/system in diagnostic tasks. In this paper we resort to DBN, a BN extension having a discrete temporal dimension. DBNs have already been applied in the domain of critical infrastructure security, for example, in [30] where a DBN is used for privacy violation detection; that model is actually simpler than the case study examined in our paper. In [31] the DBN was built according to AG, as is done in our paper, and a way to compute temporal scores from CVSS is investigated; such scores are necessary as initial probability parameters of the DBN. In this paper, besides CVSS, we resort to MITRE repository and ICS-CERT, we consider a wider case study, and both predictive and diagnostic analyses are performed.

Decision networks (DN) are another BN extension, characterised by the addition of decision and value nodes, but without a temporal dimension. In [32] DNs are exploited to evaluate the security of a SCADA architecture; decision and value nodes are exploited to model countermeasures and their effects, respectively. In this paper we do not yet consider defence mechanisms, but this can be taken into account in the future work, together with the possibility to apply dynamic decision networks (DDN) [2]; i.e., DNs characterised by a temporal dimension, as in the case of DBNs.

All such approaches start from AT or AG models to show how BNs/DBNs/DNs can be derived, stressing the evidence-based analysis allowed by these models.

In our approach we follow the same model design procedure, but we adopt the event classification defined by MITRE ATT&CK (tactics, techniques, analytics) in the AG construction and in the definition of quantitative security indices (Sections 2 and 3). One of the key problems during the model construction is the quantification in terms of probability of the basic events in the model. This task can be rather difficult in the context of critical infrastructures due to the difficulty to access historical, realistic, or confidential data. In these cases, experts can be involved in the estimation of the parameters, or events can be classified according to a probability scale [22]; for instance, attack likelihood levels *trivial–moderate–difficult–unlikely* may correspond to 0.9, 0.6, 0.3, 0.1. Another solution can be the execution of attacks in a laboratory setting, to collect statistics [33]. A further possibility is to exploit the information made available by existing sharing platforms, and in our approach we set the DBN initial parameters according to the data collected from available sources (Section 2).

Historical data may be already encoded in software tools for security assessment, as in *securiCAD* [34] which allows the user to build the architecture of ICT infrastructures, from which the AG is automatically generated according to a predefined library of potential attack steps on the assets. The AG is implemented in the form of Bayesian-like networks to be evaluated by means of Monte Carlo simulation, providing the success rate as a function of the time to compromise, for every step and asset. Despite this advantage, the built-in attack steps and their logical dependencies make it difficult to validate the underlying model and the results, limited to the success rate. In our approach, we focus on more specialised attack scenarios, including both IT and OT assets. Our DBN model is based on the ATT&CK dataset and nomenclature; it exploits evidence-based analysis (instead of simulation); it distinguishes between predictive and diagnostic analysis; and it computes a set of measures (Section 3), oriented toward attack detection. Additionally, some other works investigate tools for the automated risk identification based on systematic risk assessment methodologies. For example, the blade risk manager tool [35] allows one to evaluate the compliance with reference to cyber security standards and guidelines. The tool combines the architecture information with a cyber security knowledge base containing information related to threats, undesired events, attacks and vulnerability patterns, and security controls. The set of information used is not disclosed, making the tool obscure and not extensible. Moreover, the blade risk manager input model requires cause–effect information about the system under analysis. The architecture under analysis is provided by means of tables in Microsoft Word documents. From the architecture, a very high level model of the SCADA system is obtained and analysed.

The work presented in this paper is the prosecution of [3,36] where we developed and evaluated the BN model of a power system ICT architecture composed by three levels: IT, OT, and ICS. Several evidence-based measures were computed, such as the probability to compromise IT, OT, or ICS, the probability that a technique has been used given the compromised system or the activation of an analytic, and relevance measures for techniques and analytics. However, these measures were computed on a BN which considered the system in a unique instance of time. In this paper we examine a detailed ICS architecture for DERs, but we resort to DBN in order to model and evaluate the temporal evolution of techniques, analytics, and system state, by still exploiting the evidence-based analysis.

1.3. Structure of the Paper

The paper is organised as follows: In Section 2 we introduce our methodology for modelling the attack paths, quantitative parameter elicitation, and attack assessment. Section 3 presents and discusses experimental results obtained by the predictive and diagnostic analysis of several attack scenarios targeting our reference OT architecture, with specific attention to the DER control functionality. Finally, Section 4 concludes the paper.

2. Methodology

Our goal is the development of an analysis tool with a strong practical drive to provide guidance to security experts in assessment and planning of security measures, and in monitoring and detection of adversarial behaviours.

We base our methodology on (non exhaustive) real world data provided by the MITRE Corporation in its IT and ICS ATT&CK frameworks. The knowledge bases are compiled using publicly available data on attacks. Data are collected manually analysing reports written in English; attack processes are split into basic steps (called *techniques*) which are then classified according to the general goal they aim to achieve (the *tactic*). Although the MITRE knowledge base is not complete, the ATT&CK corpus of data is invaluable for the head start of a diagnostic and predictive analysis of cyber systems; this in turn will give a better understanding of the context and guide the collection of further data, which, later integrated in the methodology, will lead to a more specific and informed approach.

We model attack processes as sequences of techniques organised as paths of an attack graph (AG)—a directed multigraph in which nodes represent states (capabilities that an attacker has attained), and edges are attack steps (techniques). We envisage an AG modelling the entire ATT&CK matrices extended to and specialised for the energy control environment, enriched with the causal relationships that are not explicitly expressed in the matrix. Such a graph would provide a complete model of possible adversarial behaviours. This model is static; it represents prerequisites for the application of a technique and intermediate goals achieved by attackers, but does not take into account the evolution of the process in time. It is, in our approach, a first modelling step that is then instantiated, and completed with temporal capabilities, in a DBN. The DBN is a probabilistic graphical model where a directed acyclic graph is used to connect discrete random variables (nodes) to depict their dependency relation (as in BNs), with the additional concept of *time slice* that models a specific time instance. All independent nodes are parametrised with a priori probabilities that describe the likelihood of occurrence of the event they model, and all dependencies are parametrised by conditional probabilities, enabling the analysis of the evolution of the state of each single node over time. For example, if the attacker engages in exploiting a technique at a certain instant in time, this may result in the compromise of some part of the system at a subsequent instant. To model aspects related to detection, we include in our DBN also *security analytics* from the MITRE CAR: security analytics describe events whose observation is significant from a security perspective. We propose some energy sector-specific analytics that are based on application knowledge and related to the selected DER control scenario. We define the DBN model in Section 2.4 and in Section 3.1 we explain how it can be used for giving insight into the security posture of a given OT architecture, offering support for assessment, detection, and forecasting.

2.1. MITRE Knowledge Bases

The information of the IT ATT&CK knowledge base [37,38] and its ICS version [39,40] is presented in matrices whose columns are labelled by *tactics*, whereas the entries in each column are *techniques*. A tactic can be viewed as a goal that the attacker is trying to attain (e.g., persistence, impact, impair process control, etc.); a technique defines the means by which the attacker tries to attain her goal; a technique may enable more than one tactic. Each technique is documented with a description, recommendations for mitigation and detection, a list of references, a list of software that make use

of it, and a list of groups that have applied the technique. MITRE also proposes a way of scoring the “popularity” of techniques.

The ICS ATT&CK matrix appeared very recently. For the objectives of our work it presents some limitations. To start with, it is based on a smaller knowledge base: while the actor groups for the enterprise matrix are 94 and the software tools amount to 414, in the case of ICS, only 10 groups and 17 software tools are publicly known. In addition, the ICS ATT&CK project is not specific for the energy sector, and therefore it considers categories of techniques that in some cases lack granularity. Finally, it does not contemplate detection but only focuses on mitigation [39], limiting its usefulness in our case. Hence, the ICS ATT&CK matrix has been of inspiration for us, but we had to go beyond it, both in considering more specific techniques and in parametrising our models.

Most of the techniques we consider in this research are either from the ICS ATT&CK matrix or specialised from similar techniques. Only one of them is taken from the IT ATT&CK project: it is indeed one basic IT technique (new service) which we view as specialised for the ICS environment (new ICS service) to secure a foothold in a compromised node (i.e., for the tactic *persistence*).

2.2. Attack Graphs

As already mentioned, a node in an AG represents capabilities that the attacker has attained, or equivalently, the security state of the system. Edges, on the other hand, are labelled by tactic/technique pairs specifying an attack step taken by the attacker as a means for achieving a precise goal, which causes an evolution in the security state of the system (typically a downgrade).

We assume in this work that the attacker has already compromised a plant controller in the OT network by means of the external interface. We disregard here the detailed IT oriented phases of the attack (the BN model of the preliminary phases is presented in [36]). The attacker can disturb the physical process by means of a fake operation in the DER control loop.

The AG is depicted in Figure 2. The starting node is “A controls SCADA” where A is the attacker; the attacker’s target is the status “Unstable power system” (shaded node in Figure 2). As anticipated, we restrict our proof of concept analysis to a few possible attack paths. We take inspiration from [40]; for the techniques we use to label edges, but specialise and enrich them with the energy control background. We also introduce attack patterns to automatic controls of some subsystem (dotted lines in the AG of Figure 2). The subsystem is either compromised or fed by maliciously crafted data, and its reaction would disrupt operation or damage assets. In order to make the setting more realistic, we assume that some kind of check/protection is in place, forbidding in some cases such disruptive or damaging control (dotted arcs *wrong logic execution* and *correct reaction to wrong data*).

The attacker can modify the control logic of the field devices from the SCADA server, thereby getting the devices to react in unexpected ways to the commands received. Notice that this is similar to the final technique used by Stuxnet, which modified the Siemens S7 PLC software.

An alternative way to obtain the same goal is a man-in-the-middle (MITM) attack on the compromised node, intercepting data to and from the SCADA server; i.e., the plant controller. The technique *Local* MITM assumes that the attacker installs MITM malware directly on the SCADA system. A MITM technique is listed in the ICS ATT&CK matrix but that technique assumes that the attacker has not gained control of the SCADA system and either uses the address resolution protocol (ARP) spoofing or a proxy to achieve her goal. The reason for using here a more specific version of this technique is to enable a precise set of analytics to expose the attack (see Section 2.3). Being able to manage the control communications, the attacker can alter or spoof reporting messages containing the electrical measures from the field that the SCADA software in the plant controller will consider genuine and react to, causing malfunctioning. Alternatively, the MITM malware can modify outgoing command messages (technique *impact/manipulation of control*) from the SCADA system to the controlled nodes (i.e., the DER). The technique—modify reporting message—is not among those analysed in the ICS ATT&CK matrix, but is one of our energy-specific extensions.

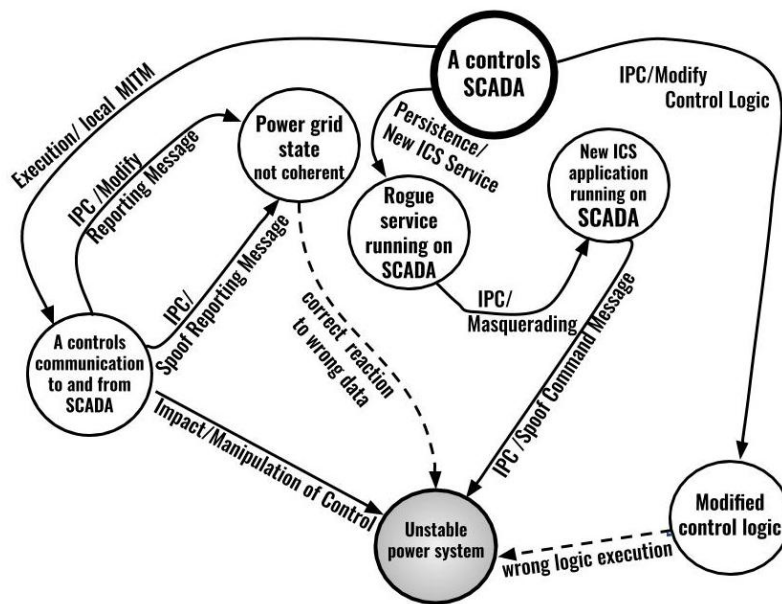


Figure 2. The attack graph.

A third attack path is the installation of a new ICS service on the SCADA victim (*persistence/new ICS service*) that pretends to be the genuine service (*IPC/masquerading*). The malicious service is protocol-specific (e.g., IEC 61850, IEC 60870-5-104, Modbus, etc.) and will send malicious commands to disrupt services or destruct assets (*IPC/spoof command message*). This path is inspired by the Ukraine attacks on the power grid on December 17, 2016 where CrashOverride/industrial malware affected the Kiev power grid area. Notice that the technique *new ICS service* does not appear in the ICS ATT&CK matrix and is a specialisation to the power domain of the analogous one from the IT ATT&CK matrix.

Notice in Figure 2 how some AG nodes impact the SCADA system: the initial compromise state, the existence of a rogue server and its masquerading as the genuine one, and the deception when the system receives spoofed or altered messages from field devices. On the other hand, when the attacker has attained the ability of modifying or spoofing messages from the field devices, her focus is on the communications. In the state in which the attacker has managed to alter the logic of the field controllers, the focus of the attack has moved to the power field. The final goal of causing instability of the power system and preventing normal operation, has a physical impact on DER behaviours.

2.3. Security Analytics

In order to focus on attack monitoring and detection, we integrated in our model analytics from the Cyber Analytics Repository (CAR) [41]. A security analytic describes events whose observation is significant from a security prospective. Each analytic is connected to the techniques that the analytic helps to expose. The ICS ATT&CK matrix has not yet been complemented with analytics. Thus, we use analytics proposed by MITRE for exposing the techniques of the ICS ATT&CK matrix wherever this is appropriate, whereas, for all techniques peculiar to the ICS world or to the power domain, we propose specific ones.

The analytics can be implemented by sensors residing on different nodes of the infrastructure. Figure 3 highlights the architectural levels that are monitored, i.e., plant controller and field devices. This will be clearer as we proceed in the presentation. Some analytics refer to incongruities in message exchanges—for example, checking the coherence with past measures and commands, or device characteristics. Others are specific for system level anomalies regarding command and application execution or location.

To expose the deployment of *modify control logic*, one must implement, for instance, software integrity checks and update monitoring (analytic UPDATE MONITORING/INSTALLED SOFTWARE

INTEGRITY CHECK). This analytic reports on the integrity of field devices. The arrow from *modify control logic* to the analytics in Figure 3 indicates that the technique exploitation by an attacker may trigger the analytics.

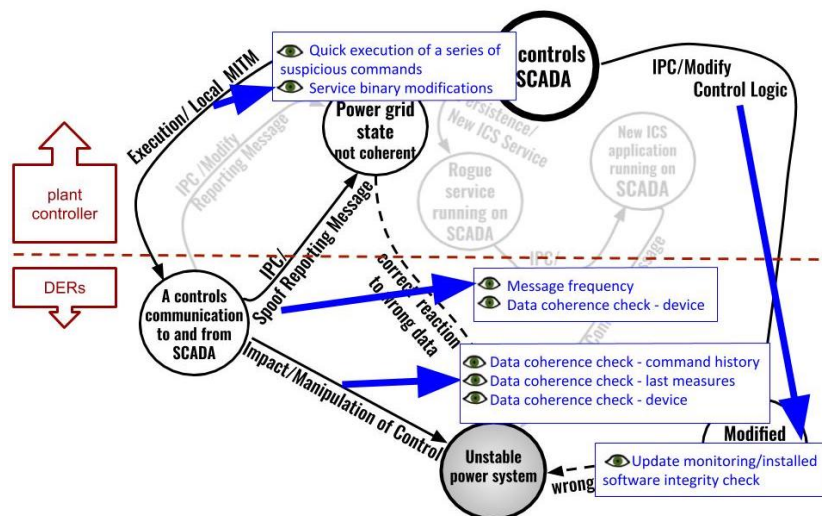


Figure 3. Example of mapping security analytics to tactics/techniques.

Local MITM can be viewed as a specialisation of IT techniques, so in this case we borrowed analytics from the IT CAR; namely, quick execution of a series of suspicious commands and service binary modifications. Both analytics report on SCADA integrity (see Figure 3): for the first one, a sensor looks for a series of specific commands executed in a quick sequence; for the latter, integrity of the SCADA binary code is verified. See in Figure 3, the correspondence.

In the case of techniques that modify or spoof command messages to disrupt normal operation, we assume that syntactic message level checks are in place and we additionally work at application level (a detection system is more effective if it has deeper knowledge of the system it defends). At a syntactical level, the check could be performed considering the information flow profile of control protocols; e.g., IEC 61850 [42] or IEC 60870-5-104 [43] message frequency. This suggests an analytic to expose the usage of the technique *spoof report message*: under the hypothesis that measure reports are sent with periodical cadence, a spoofed message will break this periodicity and therefore raise an alert. This is modelled by the analytic MESSAGE FREQUENCY. This analytic monitors correctness of communications, as depicted in Figure 3. At a semantic level, the application monitoring system would raise an alert if, e.g., a command was not consistent with the technical characteristics of the device, DATA COHERENCE CHECK–DEVICE. The latter can be implemented also to expose spoofed or modified reported data, by checking that the reports are consistent with the device characteristics. See in Figure 3 the correspondence of this analytic with techniques.

2.4. Dynamic Bayesian Networks

BNs are defined by a directed acyclic graph where nodes correspond to discrete random variables having a conditional dependence on the parent nodes. DBNs extend BNs by providing an explicit discrete temporal dimension. In this work context, the temporal dimension is introduced for capturing the time dependencies of attack steps and their effects on the operating system’s behaviour.

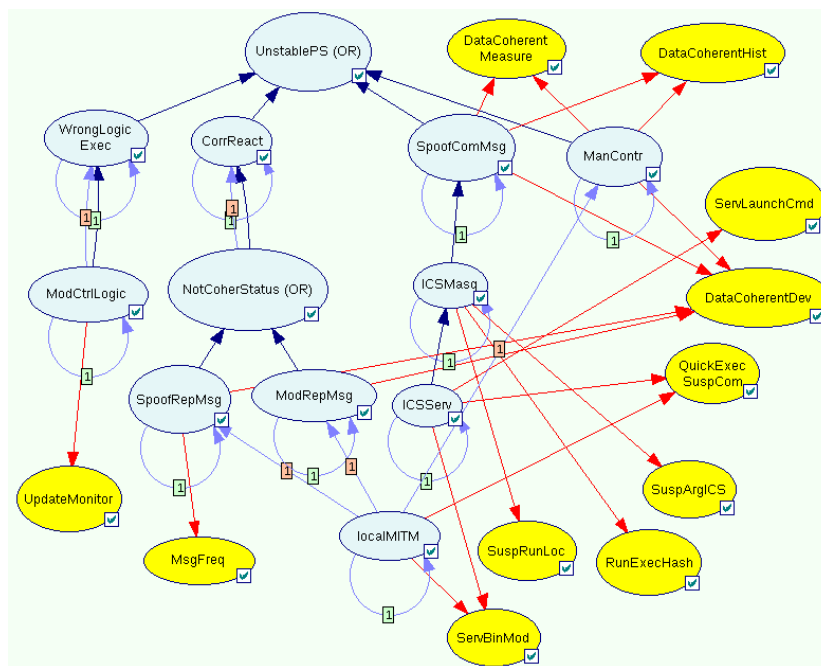
Definition 1 (Dynamic Bayesian Network). *Given a set of time-dependent state variables $X_1 \dots X_n$ and given a BN N defined on such variables, a DBN is essentially a replication of N over two time slices $t - \Delta$ and t (Δ being the so-called time discretisation step), with the addition of a set of arcs representing the transition model. Let X_i^t denote the copy of variable X_i at time slice t ; the transition model is defined through a distribution*

$P[X_i^t | X_i^{t-\Delta}, Y^{t-\Delta}, Y^t]$ where $Y^{t-\Delta}$ is any set of variables at slice $t - \Delta$ different from X_i (possibly the empty set), and Y^t is any set of variables at slice t different from X_i (possibly the empty set).

The DBN derived from the AG (Figure 2) and related analytics are shown in Figure 4 where each variable (technique, state, or analytic) is binary: the value (state) 0 means “did not occur”; 1 means “occurred.”

An arc connecting two variables (nodes) means that the state of a variable influences the state of another variable. For example, the arc going from *localMITM* to *ManContr* is necessary because the technique *manipulation of control* may occur only after the success of the technique *Local MITM*. The state of a variable in a time-slice is influenced by the state of the same variable in the previous time-slice. As an example, the arc connecting *localMITM* to itself represents the temporal evolution of this technique: if *localMITM* was inactive in the previous time-slice, it may occur in the current time-slice with a given probability. In Figure 4 all arcs labelled by a tag with the indication of the time step (always 1 in the figure) are inter-slice arcs.

The arc from *localMITM* to *ServBinMod* expresses that the analytic SERVER BINARY MODIFICATION is activated by the technique *local MITM*; In this case, we deal with intra-slice arcs establishing influences holding inside a time-slice.



Techniques, events, states		Analytics	
short name	complete name	short name	complete name
ModCtrlLogic	Modified Control Logic	UpdateMonitor	Update Monitoring
localMITM	Local Man in the Middle	ServLaunchCom	Services Launching Commands
SpoofRepMsg	Spoof Report Message	ServBinMod	Service Binary Modifications
ModRepMsg	Modified Report Message	SuspRunLoc	Suspicious Run Locations
ICSServ	ICS Service	RunExecHash	Running Executables with same
ICSMasq	ICS service masquerading		Hash and different names
SpoofComMsg	Spoof Command Message	SuspArgICS	Suspicious Arguments
ManContr	Manipulated Control	QuickExecSuspCom	Quick Execution of a series
WrongLogicExec	WrongLogicExecution		of Suspicious Commands
CorrReact	Correct Reaction	MsgFreq	Message Frequency
NotCoherStatus	Not Coherent Status	DataCoherentDev	Data Coherence – Device
UnstablePS	Unstable Power System	DataCoherentHist	Data Coherence – command History
		DataCoherentMeasure	Data Coherence–last Measures

Figure 4. DBN model of ICS attack.

The dependencies of a node are quantified in terms of conditional probabilities and are stored in its *conditional probability table* (CPT). The probability in every table entry has to be set according to the state of the parent nodes (possibly including the copy of the node in the previous time slice). As an example, Table 1 shows the CPT of the variable *SpoofRepMsg* influenced by itself and its parent node *localMITM*. The technique *SpoofRepMsg* may occur only if *localMITM* is active. Thus, for entries 1–4, the probability that *SpoofRepMsg* is active (state 1) is null because *localMITM* is inactive (state 0). For entries 5 and 6, we have that *SpoofRepMsg* was inactive at time $t - 1$ and *localMITM* occurred; therefore, *SpoofRepMsg* may happen at time t with probability 0.017156 (and may not happen with the opposite probability). For entries 7 and 8, we have that *SpoofRepMsg* was active at time $t - 1$ and *localMITM* still works; the probability that *SpoofRepMsg* is still active at time t is 1 because we assume that a technique maintains its state once activated.

Table 1. Conditional probability table of the variable *SpoofRepMsg*.

Entry	<i>localMITM</i>	($t - 1$)	(t)	prob.
1	0	0	0	1
2	0	0	1	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	0.982844
6	1	0	1	0.017156
7	1	1	0	0
8	1	1	1	1

The parameters for the techniques *new ICS service* and *masquerading* were obtained from the scores derived from the MITRE IT ATT&CK matrix. These techniques are applied to the ICS world, but they are identical to their IT counterparts, thereby justifying our choice.

For all techniques that are proper in the ICS world, we derived our scores from the US Department of Homeland Security’s ICS-CERT Advisories [5]. Precisely, from ICS-CERT we selected advisories on software that is used in the electricity sector, limiting the analysis to advisories released or revised in 2015 or later. For each of them, we classified the listed vulnerabilities according to the (ICS ATT&CK’s) technique that they enable: we inferred the latter from the vulnerability overview in the advisory. We restricted our attention to the ICS techniques in our attack graph, and furthermore selected only those advisories for which a common vulnerability scoring system (CVSS) [6] score was provided.

3. Case Study Analysis

3.1. Inference Tasks

DBNs make possible several kinds of analyses, using different types of inference algorithms. In particular, let X^t be variables at time t representing either states of the system or the fact that a technique has been exploited by the attacker. Analytics provide a stream of observations over time, which is formally denoted as $y_{a:b}$ when limited between time point a and time point b , $a \leq b$, and they consist of a set of instantiated variables Y_i^j where i denotes the specific analytic and j ($a \leq j \leq b$) the time instance. The following tasks can be performed on a DBN:

- **Monitoring** can support the security analyst or system in online diagnosis and early detection, providing a means to compute in real time the chance that an attack attempt is taking place, according to the observations gathered by the analytics. It is enabled by the **filtering** approach that consists of computing $P(X^t|y_{1:t})$, i.e., the probability of event X at time t , given the observations y_j at all instants up to t . This means tracking the probability of the system states, taking into

account the stream of observations (see Figure 5a). When this probability value exceeds a given threshold, the analyst/system can infer that an attack attempt is in progress.

- **Prediction** can support the analyst/system in identifying, on the basis of evidence obtained from the analytics, the subset of techniques that the attacker will more likely exploit in the future, thereby enabling the setup of defensive actions. The filtering approach is used in this case for computing $P(X^{t+h}|y_{1:t})$ for some horizon $h > 0$ (see Figure 5b), i.e., predicting a future state, taking into consideration evidence gathered up to now (filtering is a special case of prediction with $h = 0$).
- **Offline diagnosis** allows determining the probable causes of security events, either online, using filtering, or offline with **smoothing**. The latter consists of computing $P(X^{t-\tau}|y_{1:t})$ for some $\tau < t$, i.e., estimating what happened τ steps in the past given all the evidence (observations) up to now (see Figure 5c). Therefore, while filtering enables diagnosis in real time, smoothing makes it possible to conduct a deeper *post-incident* investigation based on evidence collected by the monitoring system during the DER operation.

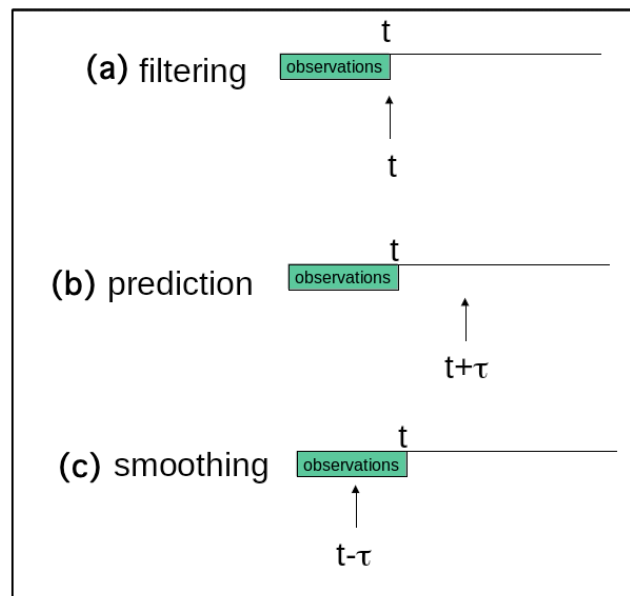


Figure 5. DBN inference tasks: filtering (monitoring), prediction, and smoothing.

In this paper we focus on the monitoring (and online diagnosis) and prediction tasks. Different algorithms, either exact (i.e., computing exact probability values) or approximate, can be exploited in order to implement the above tasks. In this paper, to design and evaluate the DBN, we use the *Bayes Net Toolbox for Matlab* (BNT) [44], and in particular, we resort to the *junction tree* (JT) [2] algorithm for the exact inference of the DBN. All the following experiments were performed on a laptop with an Intel Core i7-8750h CPU at 2.20GHz x 12 and 16 GB RAM with completion times of a few minutes.

3.2. Inference Results

We report simple analyses on our proof-of-concept attack model in order to demonstrate the usefulness and flexibility of our methodology in support of detection and defence strategies. Our model considers imperfect analytics to render the real situation in which the deployment of a technique may go undetected (false negative), or viceversa, an alert which is raised when there is no attack (false positive): we assume a probability of 10^{-4} of both false positives and false negatives. The parameter expresses the confidence that the security analyst/system has in the monitoring system; we chose it to be two orders of magnitudes smaller than the smallest probability associated to any

technique, in order to emphasise high confidence. All the experiments are carried out assuming a discrete time interval of $T = 50$ time units and the results are computed at each time step. In all figures the t values on the x -axis refer to the discretisation steps.

In our first experiment we compute the likelihood that each technique is successfully exploited by the attacker, in the setting in which no monitoring system is deployed into the DER architecture. We use the results of this experiment as a benchmark to appreciate the advantage of including the monitoring system in the architecture. It also gives us insights into the way the structural properties of the underlying attack graph affect probabilities, when not influenced by evidence.

We notice that, as expected, the most likely attack techniques are those that can be exploited by the attacker from the initial state (A controls SCADA) (see Figure 6a,e,d) and that there is an inverse correlation between the success probability of the other techniques at a fixed time t and their distance from the initial state, since the success of each step depends on the success of steps that come earlier in the attack graph. See Appendix A.1 for the details.

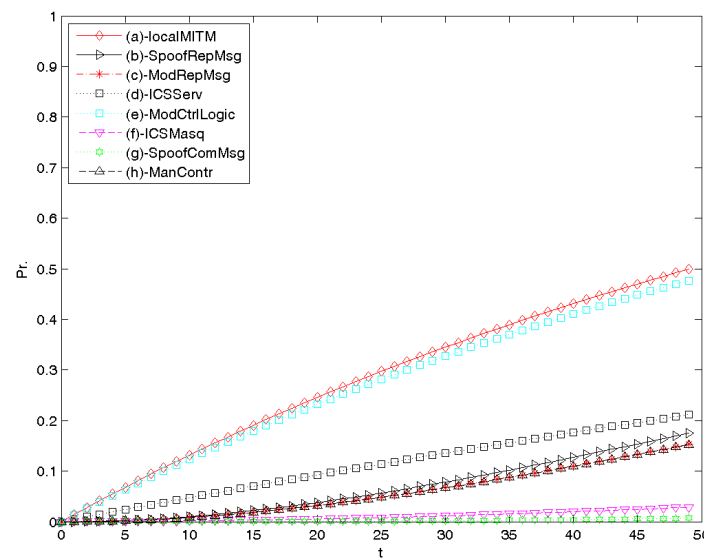


Figure 6. Probabilities of successful technique exploitation without monitoring system.

3.3. Monitoring and Online Diagnosis

As we move to the advanced setting in which a monitoring system is deployed in the DER architecture, we compare the probability that the attacker manages to reach her final goal of making the system unstable, causing a physical impact on the DER operation (the shaded node in Figure 2) in the previous setting (no monitoring system), with the probability for the same event in the setting with the monitoring system implemented. We assume that in the latter setting all analytics certify that there is no adversarial activity up to instant $t = 20$; at that point the analytic UPDATE MONITORING/INSTALLED SOFTWARE INTEGRITY CHECK starts to signal that the attacker has modified the DER control logic. All other analytics are still giving no alarm.

Figure 7 shows that, with no monitoring system in place, the chance that the attacker manages to make the system unstable, causing a physical impact on the DER behaviours, is overestimated before $t = 20$, and, even worse, after that point it is underestimated. This is a consequence of lack of information. A more detailed analysis can be found in Appendix A.2.

Now suppose that an intruder aims to attack the plant controller and its internal communications following the path by first exploiting a local MITM attack and then spoofing a DER report message containing power measures (*spoof reporting message*), leading the SCADA system to a non coherent status. This can cause a correct reaction to wrong data, which will make the whole system unstable by means of physical impact on the DER behaviours. We assume that both the analytics for local MITM (SERVICE BINARY MODIFICATIONS and QUICK EXECUTION OF A SERIES OF SUSPICIOUS COMMANDS; see

Figure 3) activate at $t = 15$, followed at $t = 25$ by the activation of the two analytics for the technique *spoof reporting message* (DATA COHERENCE CHECK—DEVICE and MESSAGE FREQUENCY; see Figure 3), whereas the other analytics do not signal any adversarial activity.

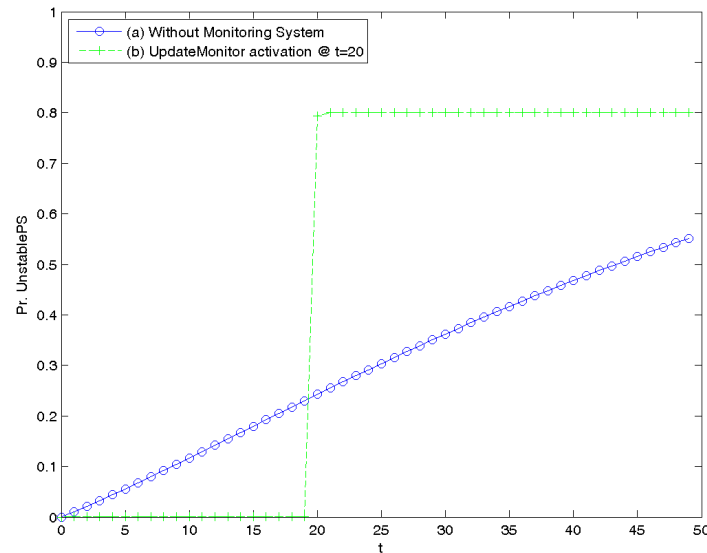


Figure 7. Probability of unstable power system: (a) without security monitoring system; (b) with security monitoring system and the observation that the UPDATE MONITORING/INSTALLED SOFTWARE INTEGRITY CHECK analytic activates at $t = 20$ (i.e., before time t the analytic is off, from t until the end is on), all other analytics are off.

Figure 8 shows that the model diagnosticates that the attacker is taking the attack path that goes through a *localMITM* and *spoof reporting message*, expecting with high probability a *correct reaction to wrong data* (see Appendix A.3). All other attacks paths are very unlikely. This type of result provides information to the cyber security team, or the automatic system, checking the status of the plant controller architecture regarding the progress of an ongoing attack inside the infrastructure in terms of the more probable attack path followed by the attacker.

The next experiment investigates the information given by our tool when analytics raise alerts in the expected order or in reverse order. To make this precise, consider the local MITM attack step and the spoofing of a DER reporting message. The latter can take place only after the plant controller local MITM has been successfully carried out by the attacker (see Figure 2). Then, if the attacker follows this attack path, the security analyst/system expects to see first analytics exposing a local MITM at plant controller and then later on analytics indicating a spoofing attack to DER communications.

Notice that for each technique only one of two analytics specific for that technique is raising an alarm. This leaves some level of indeterminacy as to whether the techniques have actually been exploited.

Figure 9 shows that out of order alerts are considered by the model less alarming, allowing for malfunctioning of the system; or are not fully credited, assuming that the attacker is following an alternative path (see Appendix A.4). While we find this response desirable, we feel that alerts that cannot be satisfactorily explained by the model should be presented as anomalies that must be investigated. This will be the object of future work.

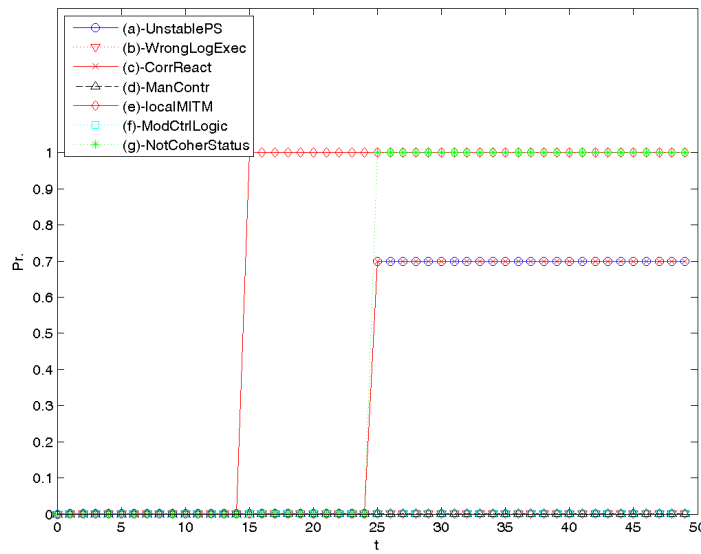


Figure 8. Task: monitoring. All analytics register no adversarial activity except for SERVICE BINARY MODIFICATIONS and QUICK EXECUTION OF A SERIES OF SUSPICIOUS COMMANDS starting at time $t = 15$, and DATA COHERENCE CHECK—DEVICE and MESSAGE FREQUENCY starting from time $t = 25$. The technique (e) (resp. the event (g)) has probability 1 starting at $t = 15$ (resp. $t = 25$). At $t = 25$ both (a) and (c) jump to probability 0.7. For (b,d,f) the probabilities of success remain negligible.

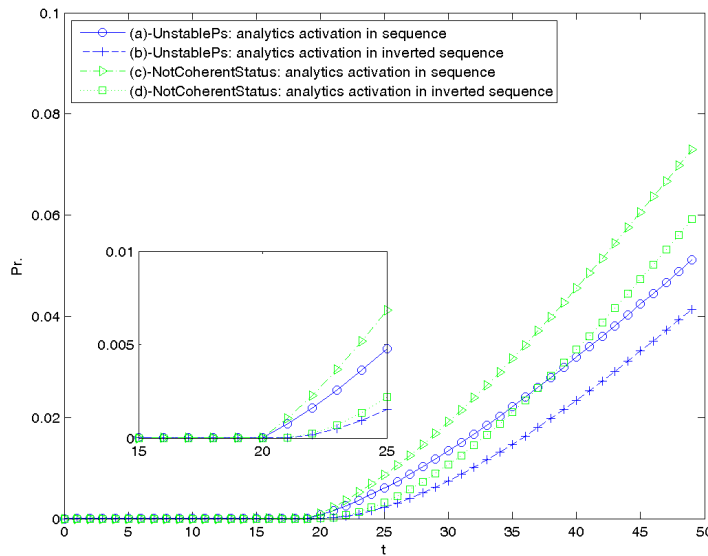


Figure 9. Task: monitoring. Analytics raising alerts in the order compatible with the attack path or in reverse order.

3.4. Prediction

The following experiments test the prediction capabilities of the model. Figures 10, 12, and 13 assume an adversarial behaviour analogous to the experiment of Figure 8; precisely at time $t = 15$ the attacker carries out a local MITM attack at plant controller and at time $t = 25$ spoofs DER reporting messages. Assuming that a full monitoring system is implemented, we analyse what predictions can be inferred from observations “up to now” when the analyst/system is making the predictions at times $t_p = 10$, $t_p = 20$ and $t_p = 30$ for all successive instants. The ability to anticipate the attackers’ behaviour is surely a key function for improving the fast response to security anomalies in a continuously evolving threat landscape.

When the prediction is made at time $t_p = 10$, no analytic has raised an alert, and therefore up to that moment monitoring enables us to conclude that no attack was in progress. Figure 10 shows that the probability of occurrence of all events monitored was negligible until the present time ($t_p = 10$). Afterwards the slopes of all curves are identical to the slopes in Figure 11 (which reports the probabilities computed when no monitoring system is deployed), since the security analyst/system has no monitoring information for the future and cannot rely on the absence of attacks so far to feel safe in the future (cf. Appendix A.5).

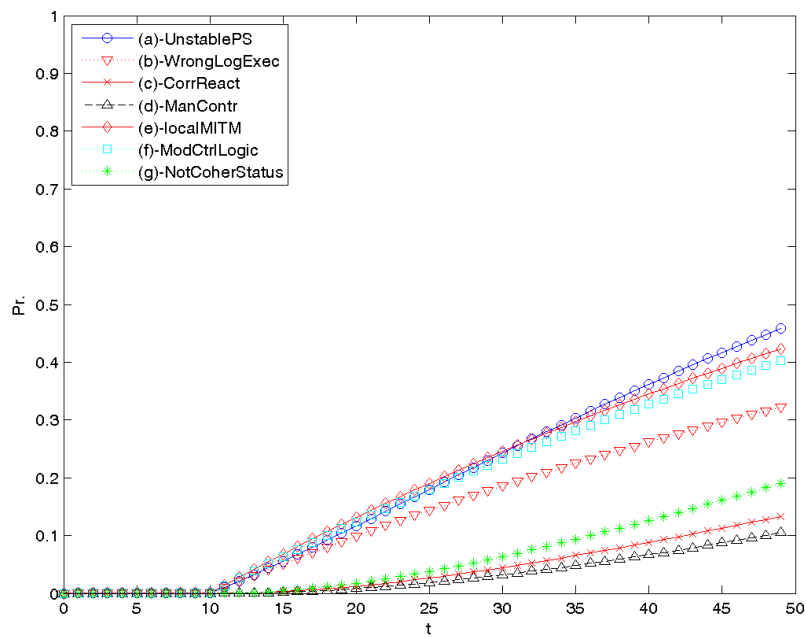


Figure 10. Task: prediction. At time $t = 10$ on the basis of the observations up to time t , predictions for all subsequent times are computed.

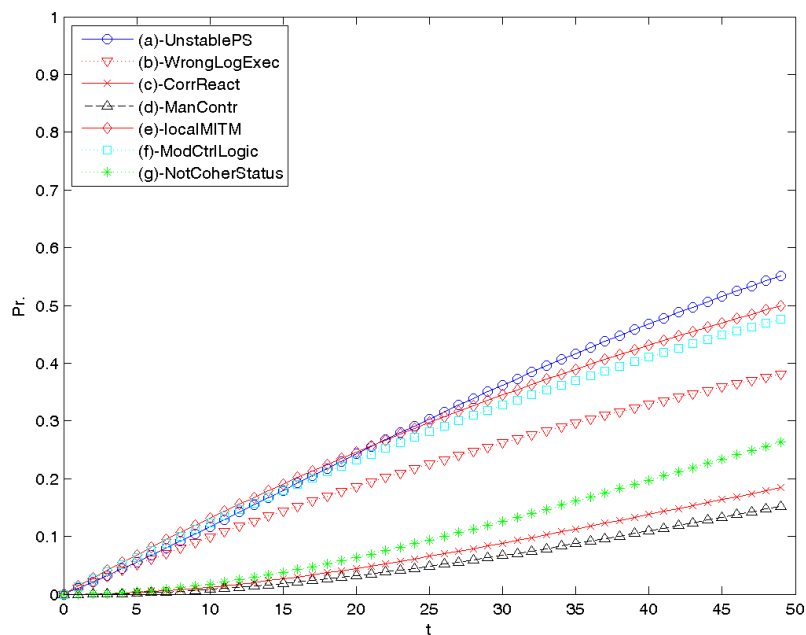


Figure 11. Task: monitoring. For comparison with Figure 10 probabilities of the same events in case of no evidence are plotted.

Figure 12 assumes that the security analyst/system is monitoring the infrastructure at time 20 (prediction at time $t_p = 20$). Before $t_p = 20$ the analytics monitoring, a local MITM attack at plant controller has raised an alarm at time 15. Then, at time $t_p = 20$ it is already known that the attacker has taken that first step. This knowledge has an influence on the prediction (as is discussed in detail in Appendix A.5), which shows the significance of the model, since it gives the analyst/system a tool to forecast adversarial moves based on previous observations.

Finally, Figure 13 shows the results of a prediction made at $t_p = 30$. Up to $t = 30$ all evidence from the monitoring system is available, and therefore the information is the same as in the monitoring task of Figure 7. After that, the analyst/system no longer has monitoring evidence, and therefore must resort to a prediction. It can be seen that the probability that the system is made unstable, deviating the DER behaviours, is now significantly higher since more evidence has been gathered, and in addition, the model leaves open the possibility that the attacker might also follow a second attack path in the future. For more details, cf. Appendix A.5.

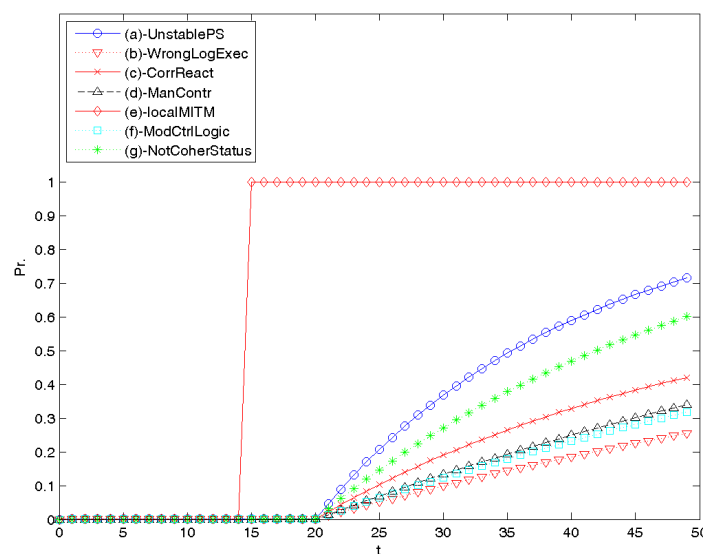


Figure 12. Task: prediction. At time $t = 20$ on the basis of the observations up to time t , predictions for all subsequent times are computed.

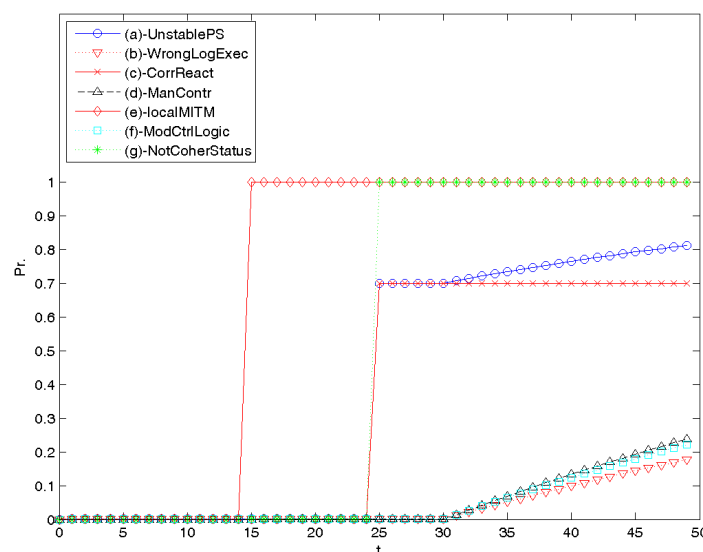


Figure 13. Task: prediction. At time $t = 30$ on the basis of the observations up to time t , predictions for all subsequent times are computed.

4. Discussion and Conclusions

The cyber resilience of digital energy systems requires advanced methodologies and tools for responding proactively to the evolution of cyber threats. Our research evaluates multistep attack processes modelled by means of DBNs enabling predictive and diagnostic analyses. Specialised for the power domain and based on real world data, the methodology is a promising approach to human-driven and automated support for advanced defence strategies.

The monitoring capabilities allow the security analyst/system to have a real time snapshot of the current status of the ICT infrastructure: the monitoring system highlights a critical point and permits the activation of real time countermeasures. The prediction functionality allows one to anticipate the state of the ICT infrastructure, and to activate defensive measures in order to prevent the escalation of an ongoing attack process. The approach robustness guarantees that the detection system is able to raise alerts also in case of unexpected attacker behaviour or monitoring anomalies, as observed in presented experiments (see Figure 9). The strength of the approach is, also, the ability to include knowledge coming from the real world and to provide cyber security analyses specific for the energy context.

In this paper we have modelled a high confidence in the analytics by the security analyst/system. The impact on the results is that they do not reveal appropriately some unforeseen evolutions: the attacker following a different path, non-adversarial causes triggering an alert, or even some malfunctioning or misconfiguration of the monitoring system itself. As is, the model can be used for correlating selected analytics with covered attack processes. A next step will be a relaxation of these hypotheses allowing higher probabilities of error of the analytics; this way the system would not dismiss an alert on the basis that another analytic has not raised an alarm as expected. The uncertainty of the analytics, though, must be fine tuned so that the monitoring system is still reliable enough to be informative. On the other hand, we also plan in future work to directly monitor the consistency of alerts received from the monitoring system.

From a security analyst/system perspective, having a model with an interpretable structure (see Section 1.2) is desirable in order to gain enough trust in the conclusions drawn by the modelling tool to really support a potentially costly decision. For instance, in an extreme situation, when having such a high confidence on the security monitoring and prediction tool, it might be preferable to detach a threatened DER, thereby incurring energy production losses, rather than running the risk of cascading damages due to its unpredictable behaviour.

The interpretable structure is relevant also with reference to the NIS directive, which requires service operators to report on cyber incidents. In this context, our methodology facilitates the accurate reporting, by its ability to explain the attack processes from collected evidence.

As we anticipated in the introduction, we plan to refine the methodology with more details about attack models and analytics, and to extend the analysis tool with the actual implementation of such analytics in a laboratory setup. Additionally, it should be kept in mind that the AG and the DBN presented are proofs of concept whose aim is to demonstrate capabilities and limits of our approach, and to illustrate the steps we intend to take in the future. The AG and DBN that will be validated in the test laboratory first (and eventually in the operational environment) are going to cover all techniques proposed or inspired by the MITRE ATT&CK matrices and by the CAR project, as is useful for the electric domain. Finally, we expect that the validation activity will lead to identification of additional techniques and analytics to be included in the current model. Moreover the model can be continuously extended by following the threat evolution.

Author Contributions: All the authors contributed to every section of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work is original and has been supported by a joint collaboration between RSE S.p.A. and Università del Piemonte Orientale, funded by the Research Fund for the Italian Electrical System in compliance with the Decree of Minister of Economic Development April 16, 2018, and partially by Università del Piemonte Orientale.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

AG	Attack Graph	ICS	Industrial Control System
ARP	Address Resolution Protocol	ICT	Information and Communication Technology
AT	Attack Tree	IoT	Internet of Things
BN	Bayesian Network	IPC	Impair Process Control
CAR	Cyber Analytics Repository	IT	Information Technology
CVSS	Common Vulnerability Scoring System	MITM	Man in the Middle
DBN	Dynamic Bayesian Network	OT	Operational Technology
DER	Distributed Energy Resource	SCADA	Supervisory Control and Data Acquisition
DMZ	Demilitarised Zone	SGU	Significant Grid User
DSO	Distribution System Operator	TSO	Transmission System Operator

Appendix A. Detailed Analysis of Experimental Results

Appendix A.1. No Evidence

In this experiment we compute the likelihood that each technique is successfully exploited by the attacker, in the setting in which no monitoring system is deployed into the DER architecture.

Figure 6 shows the cumulative probabilities of the technique success. Initially, it is more likely that the attacker launches a local MITM attack while targeting the control communications (Figure 6a), tries to modify the DER control logic (*modify control logic*, Figure 6e), or installs a rogue service on the SCADA system of the plant controller (*new ICS service*, Figure 6d): this corresponds to expectations, given that these are the only steps that the attackers can carry out from their initial stance (cf. Figure 2). The difference between the success probabilities of these first three techniques depends on the probabilities assigned to them on the basis of the ATT&CK scores or CVSS values (see Section 2.4). For instance, *new ICS service* has much lower probability than the other two.

Moreover, we can observe an inverse correlation between the success probability of the other techniques at a fixed time t and their distance from the initial node (*a controls SCADA*) since the chance of success of a high-level attack step depends on the success of lower steps in the attack path. We notice that the less likely attack step is that a command message is spoofed (*spoof command message*, Figure 6g) since this attack step can be made only after

a new ICS service, including the malicious functionality is installed (for which as we noticed, a rather low probability of success has been estimated, Figure 6d), and that service successfully masquerades as a legitimate plant controller (*masquerading*, Figure 6f).

Appendix A.2. With vs. without a Monitoring System

We compare here the probability that the attacker manages to reach her final goal of making the system unstable causing a physical impact on the DER operation in the setting with no monitoring system with the probability for the same event in the setting in which the monitoring system is implemented. Figure 7b shows that, when the monitoring system is deployed, the probability of instability is negligible (in the order of 10^{-20}) before $t = 20$ since all the nearly perfect analytics are reporting no adversarial activity. When the analytic UPDATE MONITORING/INSTALLED SOFTWARE INTEGRITY CHECK signals that updates have been improperly installed or software integrity has been violated on DER control components, it is recognised that the attacker has modified the DER control logic (while all other analytics report no incidents). At that point, there is a high probability that the wrong logic is executed (if protection mechanisms fail); this probability has been estimated at 0.8. As a consequence since the system is almost sure that the attacker has succeeded in modifying the DER control logic, the system will be unstable with probability almost 0.8. In contrast, Figure 7a

shows that the chance that the attacker manages to make the system unstable, causing physical impact on the DER behaviours, is overestimated before $t = 20$, when no monitoring system is deployed, and, even worse, after that point it is underestimated. In both cases the distorted evaluation is due to the fact that no evidence is available and therefore the security analyst/system has no information on the adversarial activity.

Appendix A.3. Attack Path

In this setting an intruder aims to attack the plant controller and its internal communications following the path by first exploiting a local MITM attack and then spoofing a DER report message containing power measures (*Spoof Reporting Message*), leading the SCADA system to a non coherent status. This can cause a correct reaction to wrong data, which will make the whole system unstable by means of physical impact on the DER behaviours. We assume that both the analytics for local MITM (SERVICE BINARY MODIFICATIONS and QUICK EXECUTION OF A SERIES OF SUSPICIOUS COMMANDS; see Figure 3) activate at $t = 15$, followed at $t = 25$ by the activation of the two analytics for the technique *spoof reporting message* (DATA COHERENCE CHECK—DEVICE and MESSAGE FREQUENCY; see Figure 3), whereas the other analytics do not signal any adversarial activity.

In this setting the monitoring task (see Figure 8) alerts the security analyst/system after time $t = 15$ that a local MITM has taken place in the plant controller and its internal communications (Figure 8e), and later, after $t = 25$, that the DER electrical state is not coherent (Figure 8g), and therefore that with high probability (0.7) the system will soon be unstable (Figure 8a), since with high probability the system will react (correctly) to the wrong data (Figure 8c). In contrast, the analyst/system can verify that no other attack is in progress, since all other analytics have not raised alarms; Figure 8 shows some examples (cf. (b,d,f)). Before $t = 15$, no attacks are detected by any analytic and therefore the analyst/system can trust that no attacks are underway, only with a negligible probability of error (in the order of 10^{-30}) due to the high confidence in the monitoring system that the analyst has (and that has been modelled).

Appendix A.4. Order of Alerts

This experiment investigates the information given by our tool when analytics raise alerts in the expected order or in reverse order, specifically in the following two settings:

- Correct order: at $t = 15$, QUICK EXECUTION OF A SERIES OF SUSPICIOUS COMMANDS raises an alert (suggesting a local MITM at plant controller is in progress) and at $t = 20$ something wrong is detected in the MESSAGE FREQUENCY of DER reporting messages (which is a sign that the attacker might be trying to do some spoofing);
- Reverse order: at $t = 15$, the analytic MESSAGE FREQUENCY raises an alert, at $t = 20$ QUICK EXECUTION OF A SERIES OF SUSPICIOUS COMMANDS does.

The results in Figure 9 show that there are possible issues for the stability of the system or for the coherence of the SCADA system state only after both alerts have been raised. This is consistent with the fact that an isolated step is less alarming: if it is the local MITM at the plant controller, then it is just the beginning of a rather long attack path (Figure 9a,c); on the other hand, if the first one is the spoofing of a DER reporting message, it is considered a mistake because it cannot have taken place if its preconditions (the local MITM) have not been verified (Figure 9b,d). After $t = 20$, however, in both cases the success probability of the attacker is considered as real and increasing, with less confidence when the activation is out of sequence. The desirable aspect of this outcome is that the monitoring task can distinguish appropriately the order of the alerts in time and react accordingly; at the same time it does not fully disregard alarms that have been given in an unexpected order. On the other hand we find unsatisfactory that prior to $t = 20$ the state of alert is negligible in the case of inverted analytic activation (while it starts growing before $t = 20$ in case the order is as expected). We feel that some kind of alarm should be raised anyway, if only to warn that something might be wrong with the monitoring

system or to account for the possibility that the attacker is following an alternative, unforeseen, attack path. The alternative interpretation, that the first analytic should be disregarded since out of time, is not fully coherent with the growth of the alarm after $t = 20$. Each one of the techniques involved in this experiment has two analytics associated, but for each one only one of the two is raising an alarm. This leaves some level of indeterminacy as to whether the techniques have actually been exploited, and explains why the computed probabilities are rather low.

Appendix A.5. Predictions

The experiments testing the prediction capabilities of the model assume an adversarial behaviour analogous to the experiment of Figure 8: at time $t = 15$ the attacker carries out a local MITM attack at plant controller and at time $t = 25$ spoofs DER reporting messages. In this setting, the security analyst/system has a view of evidence up to the prediction time and must forecast adversarial moves based on this data. We carry out three experiments with predictions made at times $t_p = 10$, $t_p = 20$ and $t_p = 30$, for all successive instants.

When the prediction is made at time $t_p = 10$, no analytic has raised an alert, and therefore up to that moment, monitoring enables to conclude that no attack was in progress. This is evidenced in Figure 10 by the fact that the probability of occurrence of all events monitored is negligible. After $t_p = 10$ the analyst/system has no evidence to depend upon, since we are assuming that $t_p = 10$ is the present and therefore all instants $t > 10$ lie the future. Then all that can be said about probabilities of future attack depends on the *a priori* probabilities of techniques (that were computed as in Section 2.4). In Figure 11 we plotted the probabilities of the same events, in the setting in which no monitoring system is deployed into the DER architecture (the same experiment reported in Figure 6 but showing different events). It can be seen that the slopes of all curves after $t_p = 10$ in Figure 10 are identical to the slopes in Figure 11, since the security analyst/system has no monitoring information in the future: the fact that there have been no attacks up $t_p = 10$ cannot be used to infer security of the system afterwards, or equivalently stated, the analyst cannot relax simply because the system has not been attacked so far.

Figure 12 assumes that the security analyst/system is monitoring the infrastructure at time 20 (prediction at time $t_p = 20$). Before $t_p = 20$ more information has been collected from the monitoring system. In particular the analytics monitoring a local MITM attack at plant controller have raised an alarm at time 15. Then, at time $t_p = 20$ it is already known that the attacker has taken that first step. This knowledge has an influence on the prediction, as will be discussed in detail later, which shows the significance of the model, since it gives the analyst/system a tool to forecast adversarial moves based on previous observations. Precisely, it can be seen that the probability that the attacker manages to make the system unstable is significantly higher at time $t = 30$, say, than it was for the prediction at time $t_p = 10$ (0.3680 as opposed to 0.2425), and increases faster: for instance if we compare the probabilities predicted for 30 time units in the future ($t_p + 30$), for the prediction made at $t_p = 10$ it is less than 0.4 at time $t_p + 30 = 40$, and over 0.6 at time $t_p + 30 = 50$ for the prediction made at $t_p = 20$. The probability that the adversarial activity will be completely successful is anyway not very high, since no other attack steps have been observed. Additionally, notice that the slopes of probabilities that the attacker will try to manipulate the DER control (*Manipulation of Control*) or that the system will correctly react on wrong data (*correct reaction to wrong data*) are noticeably steeper in the prediction at time $t_p = 20$ than they were in the prediction at $t_p = 10$: this is due to the fact that both techniques have local MITM at plant controller as a precondition, and in the case $t_p = 20$ it is known that the precondition has occurred.

Finally, Figure 13 shows the results of a prediction suppose the security analyst/system at $t_p = 30$. Up to $t = 30$ all evidence from the monitoring system are available, and therefore the information is the same as in the monitoring task of Figure 7. After that, the analyst/system no longer has monitoring evidence, and therefore must resort to a prediction. In Figure 13 it can be seen that the probability that the system is made unstable deviating the DER behaviours is now significantly

higher. it corresponds to the probability that the plant controller SCADA software has a correct reaction to wrong data (because that final step would directly lead to DER behaviour deviation and so making the system unstable, as can be checked in Figure 2) but then grows even further since it must be taken into account also that the attacker might follow a different attack path. it can be observed that in Figure 13, after $t = 30$ the probabilities of all events not related to the four analytics that have raised alerts grow with the same slope as they do in Figure 12 after $t = 20$: this is consistent with the fact that in both cases the corresponding analytics had not signalled any danger prior to t_p and no further information is available afterwards.

References

1. European-Union. The Directive on Security of Network and Information Systems (NIS Directive). Available online: <https://ec.europa.eu/digital-single-market/en/network-and-information-security-nis-directive> (accessed on 15 June 2020).
2. Portinale, L.; Codetta, D. *Modeling and Analysis of Dependable Systems: A Probabilistic Graphical Model Perspective*; World Scientific: Singapore, 2015.
3. Cerotti, D.; Codetta, D.; Dondossola, G.; Egidi, L.; Franceschinis, G.; Portinale, L.; Terruggia, R. Analysis and Detection of Cyber Attack Processes targeting Smart Grids. In Proceedings of the Innovative Smart Grid Technologies Europe (ISGT), Bucharest, Romania, 29 September–2 October 2019.
4. The MITRE Corporation. General Resources MITRE ATT&CK. Available online: <https://attack.mitre.org/resources/> (accessed on 15 June 2020).
5. ICS-CERT. Advisories, 2018. Available online: <https://ics-cert.us-cert.gov/advisories> (accessed on 15 June 2020).
6. CVSS SIG. FIRST. Available online: <https://www.first.org/cvss/> (accessed on 15 June 2020).
7. Hutchins, E.; Cloppert, M.; Amin, R. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Lead. Issues Inf. Warf. Secur. Res.* **2011**, *1*, 80.
8. The MITRE Corporation. Threat-Based Defense. Available online: <https://www.mitre.org/capabilities/cybersecurity/threat-based-defense> (accessed on 15 June 2020).
9. ICS-CERT. Primary Stuxnet Advisory. Available online: <https://ics-cert.us-cert.gov/advisories/ICSA-10-272-01> (accessed on 15 June 2020).
10. ICS-CERT. Cyber-Attack Against Ukrainian Critical Infrastructure. Available online: <https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01> (accessed on 15 June 2020).
11. ICS-CERT. CRASHOVERRIDE Malware. Available online: <https://ics-cert.us-cert.gov/alerts/ICS-ALERT-17-206-01> (accessed on 15 June 2020).
12. Sun, C.C.; Hahn, A.; Liu, C.C. Cyber security of a power grid: State-of-the-art. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 45–56.
13. Dondossola, G.; Terruggia, R. Security of communications In voltage control for grids connecting Distributed Energy Resources: impact analysis and anomalous behaviours. *Cigrè Sci. Eng. Innov. Power Syst. Ind. J.* **2015**, *2*, 30–39.
14. Korman, M.; Valja, M.; Bjorkman, G.; Ekstedt, M.; Vernotte, A.; Lagerstrom, R. Analyzing the effectiveness of attack countermeasures In a SCADA system. In Proceedings of the Workshop on Cyber-Physical Security and Resilience In Smart Grids, Pittsburgh, PA, USA, 18–21 April 2017; pp. 73–78.
15. Kaur, K.J.; Hahn, A. Exploring ensemble classifiers for detecting attacks In the smart grids. In Proceedings of the Fifth Cybersecurity Symposium, Coeur d' Alene, ID, USA, 9–11 April 2018; p. 13.
16. Hassanzadeh, A.; Burkett, R. SAMIT: Spiral Attack Model In IIoT Mapping Security Alerts to Attack Life Cycle Phases. In Proceedings of the Industrial Control Systems Cyber Security Research ICS-CSR, Hamburg, Germany, 29–30 August 2018; pp. 11–20.
17. M.Touhiduzzaman.; Hahn, A.; Srivastava, A. ARCADES: Analysis of Risk from Cyber Attack against DEfensive Strategies for power grid. *IET Cyber-Phys. Syst. Theory Appl.* **2018**, *3*, 119–128, doi:10.1049/iet-cps.2017.0118.
18. Kim, J.; Park, J. FPGA-based network intrusion detection for IEC 61850-based industrial network. *ICT Express* **2018**, *4*, 1–5, doi:10.1016/j.icte.2018.01.002.

19. Yang, Y.; Xu, H.; Gao, L.; Yuan, Y.; McLaughlin, K.; Sezer, S. Multidimensional Intrusion Detection System for IEC 61850-Based SCADA Nets. *IEEE Trans. Power Deliv.* **2017**, *32*, 1068–1078, doi:10.1109/TPWRD.2016.2603339.
20. Karlsen, D.; Uhlen, K.; Vormeda, L.K. Introducing PMU-based Applications In the Control Room Setting. In Proceedings of the CIGRE—Int. Council on Large Electric Systems Session, Paris, France, 26–31 August 2018.
21. Byres, E.J.; Franz, M.; Miller, D. The use of attack trees In assessing vulnerabilities In SCADA systems. In Proceedings of the International Infrastructure Survivability Workshop, Lisbon, Portugal, 5–8 December 2004.
22. Ten, C.; Manimaran, G.; Liu, C. Cybersecurity for Critical Infrastructures: Attack and Defense Modeling. *IEEE Trans. Syst. Man, Cybern. Part Syst. Humans* **2010**, *40*, 853–865.
23. LeMay, E.; Ford, M.D.; Keefe, K.; Sanders, W.H.; Muehrcke, C. Model-based Security Metrics Using ADversary Vlew Security Evaluation (ADVISE). In Proceedings of the International Conference on Quantitative Evaluation of Systems, Aachen, Germany, 5–8 September 2011; pp. 191–200.
24. Yegnanarayana, B. *Artificial Neural Networks*; PHI Learning Pvt. Ltd.: New Delhi, India, 2009.
25. Arrieta, A.B.; Diaz-Rodriguez, N.; Del-Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115.
26. Qin, X.; Lee, W. Attack Plan Recognition and Prediction Using Causal Networks. In Proceedings of the ACSAC, Tucson, AZ, USA, 6–10 December 2004; pp. 370–379.
27. Xie, P.; Li, J.; Ou, X.; Liu, P.; Levy, R. Using Bayesian Networks for cyber-security analysis. In Proceedings of the International Conference on Dependable Systems & Networks (DSN), Chicago, IL, USA, 28 June–1 July 2010; pp. 211–220.
28. Zhang, S.; Song, S. A novel attack graph posterior inference model based on Bayesian network. *J. Inf. Sec.* **2011**, *2*, 8–27.
29. Lyu, X.; Ding, Y.; Yang, S. Bayesian Network Based C2P Risk Assessment for Cyber-Physical Systems. *IEEE Access* **2020**, *8*, 88506–88517.
30. An, X.; Jutla, D.; Cercone, N. Privacy intrusion detection using dynamic Bayesian networks. In Proceedings of the ICEC, Cambridge, UK, 20–22 September 2006; pp. 208–215.
31. M. Frigault, L. Wang, A.S.; Jajodia, S. Measuring network security using dynamic bayesian network. In Proceedings of the Workshop on Quality of protection, Alexandria, VA, USA, 27 October 2008; pp. 23–30.
32. Codetta, D.; Portinale, L.; Terruggia, R. Quantitative Evaluation of Attack/Defense Scenarios through Decision Network Modelling and Analysis. In Proceedings of the IEEE International Carnahan Conference on Security Technologies, Rome, Italy, 13–16 October 2014; pp. 432–437.
33. Dondossola, G.; Garrone, F.; Szanto, J. Cyber risk assessment of power control systems—A metrics weighed by attack experiments. In Proceedings of the IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–28 July 2011; pp. 1–9.
34. Foreseeti. securiCAD. Available online: <https://www.foreseeti.com/securicad/> (accessed on 15 June 2020).
35. Analytics, K. Blade RiskManager. Available online: <https://kdmanalytics.com/cybersecurity-products/blade-riskmanager/> (accessed on 15 June 2020).
36. Cerotti, D.; Codetta, D.; Dondossola, G.; Egidi, L.; Franceschinis, G.; Portinale, L.; Terruggia, R. A Bayesian Network Approach for the Interpretation of Cyber Attacks to Power Systems. In Proceedings of the ITASEC, Pisa, Italy, 13–15 February 2019.
37. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. *MITRE ATT&CK™: Design and Philosophy*; Technical Report; The MITRE Corporation: McLean, VA, USA, 2018.
38. The MITRE Corporation. Adversarial Tactics, Techniques and Common Knowledge (ATT&CK), 2015. Available online: <https://attack.mitre.org/> (accessed on 15 June 2020).
39. Alexander, O. Launching ATT&CK for ICS. Available online: <https://medium.com/mitre-attack/launching-attack-for-ics-2be4d2fb9b8> (accessed on 15 June 2020).
40. The MITRE Corporation. ATT&CK for Industrial Control Systems, 2020. Available online: https://collaborate.mitre.org/attackics/index.php/Main_Page (accessed on 15 June 2020).
41. The MITRE Corporation. Cyber Analytics Repository (CAR). Available online: https://car.mitre.org/wiki/Main_Page (accessed on 15 June 2020).

42. IEC 61850 International Standard—Communication Networks and Systems for Power Utility Automation. IEC Technical Committee 57—Working Group 10. IEC 61850:2020 SER. Available online: <https://webstore.iec.ch/publication/6028> (accessed on 15 June 2020).
43. IEC 60870-5-104 International Standard—Telecontrol Equipment and Systems—Part 5-104: Transmission Protocols—Network Access for IEC 60870-5-101 Using Standard Transport profiles IEC Technical Committee 57—Working Group 3. IEC 60870-5-104:2006+AMD1:2016. Available online: <https://webstore.iec.ch/publication/25035> (accessed on 15 June 2020).
44. Bayes Net Toolbox for Matlab. Available online: <https://github.com/bayesnet/bnt> (accessed on 15 June 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).