Journals
The Institution of
Engineering and Technology

Inferring Adversarial Behavior in Cyber-physical Power Systems using a Bayesian Attack Graph Approach

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Abhijeet Sahu[1], *Katherine Davis*[1]

[1] *Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA*
** E-mail: abhijeet_ntpc@tamu.edu, katedavis@tamu.edu*

Abstract: Highly-connected smart power systems are subject to increasing vulnerabilities and adversarial threats. Defenders need to proactively identify and defend new high-risk access paths of cyber-intruders that target grid resilience. However, cyber-physical risk analysis and defense in power systems often requires making assumptions on adversary behavior, and these assumptions can be wrong. Thus, this work examines the problem of inferring adversary behavior in power systems to improve risk-based defense and detection. To achieve this, a Bayesian approach for inference of the Cyber-Adversarial Power System (Bayes-CAPS) is proposed that uses Bayesian networks (BNs) to define and solve the inference problem of adversarial movement in the grid infrastructure toward targets of physical impact. Specifically, BNs are used to compute conditional probabilities to queries, such as the probability of observing an event given a set of alerts. Bayes-CAPS builds initial Bayesian attack graphs (BAGs) for realistic power system cyber-physical models. These models are adaptable using collected data from the system under study. Then, Bayes-CAPS computes the posterior probabilities of the occurrence of a security breach event in power systems. Experiments are conducted that evaluate algorithms based on time complexity, accuracy and impact of evidence, for different scales and densities of network. The performance is evaluated and compared for five realistic cyber-physical power system models of increasing size and complexities ranging from 8 to 300 substations based on computation and accuracy impacts.

1 Introduction

Power systems are cyber-physical critical infrastructure that need defense against a wide range of threats involving various adversary motivations, capabilities, and tactics. Hence, it is indispensable to prioritize grid resilience to cyber threats. The need is evidenced from major historical events, e.g., Stuxnet in 2008 [1], Ukraine in 2015 [2], and an intrusion into the European Network of Transmission System Operators for Electricity (ENTSO-E) in 2020 with potential to compromise 42 transmission system operators across 35 member states in Europe [3].

The behavioral characteristics of threat actors are important to study to assess the potential of early events to propagate to physical impact. How threats can propagate through adversarial movement in a network with vulnerability exploitation and privilege escalation is crucial knowledge for stakeholders. The way to obtain such knowledge is a cyber-security risk assessment [4]. Risk assessments are often assisted by attack trees or attack graphs, a type of graph formalism for analyzing network and host vulnerabilities in terms of access paths to compromise a target [5]. An attack graph captures the relationships among various vulnerability exploits that could be incorporated by the intruder, along with the privileges escalated, to compromise a single or a set of targets.

Despite the importance and the need, major challenges exist in inferring the point of intrusion and compromised elements. While intrusion detection systems (IDSs), system logs, and various cyber and physical side features contain evidence, the evidence only observes symptoms of an event and must be interpreted. Two types of uncertainty contribute to this challenge, i.e., *aleatory*, caused by random behavior of systems, and *epistemic*, caused by lack of complete knowledge of the system. Three types of methods from probability theory are relevant to address both types of uncertainty in intrusion analysis: a) Monte-Carlo, b) Bayesian, and c) Dempster-Shafer theory [6]. Among these, Bayesian Networks (BNs) are adopted in this work for four major reasons:

1. The BN formalism is versatile and allows ease of construction from many sources, e.g., cybersecurity domain experts' prior knowledge, different threat models, and by learning from raw data. By comparison, the *Inter-Domain Evidence theoretic Approach for Inference (IDEA-I)* based on Dempster Shafer Theory of Evidence (DSTE) [7] is shown to assist in reducing false alarms, but has challenges with incorporating domain knowledge, high computational expense, and inability to capture causal relationships between events. Compared to DSTE, BNs are advantageous for incorporating prior model information in the defense of power systems.
2. BNs are probabilistic graphical models (PGMs) that help account for uncertainty of an adversary's behavior. For instance, in the 2015 Ukraine attack, after the intruder obtained remote access to the operational technology (OT) network, it focused on a malicious firmware update to deploy in the HMI [2], while in the Colonial Pipeline attack of 2021, after a threat affiliate obtained remote access by exploiting a vulnerability (CVE-2021-20016) in SonicWall SMA100 SSL VPN, it planted a backdoor, SmokedHam, and issued a press release of the compromised state of the firm in NASDAQ and demanded ransom [8], which conveys that *from a same security state, an adversary can adopt different trajectory.*
3. The BN formalism is capable of performing the important function of causal reasoning between each step in the access paths of the adversary's trajectory to compromise the target.
4. For decision making problems, Bayesian Reinforcement Learning (BRL) is being extensively studied [9] and relies upon the accuracy of the inferences, further motivating the study of BNs for their safety and accuracy within a critical infrastructure environment.

Using BN, two related problems can be formulated and addressed: a) Bayesian inference, and b) Bayesian structure learning. This work extends [10] that considered the Bayesian structure learning problem to learn the structure of the attack graph, given the raw data, with constraint-based and score-based techniques. By comparison, this work focuses on Bayesian inference, with application to detection as well as response under uncertainty using BRL. Hence, a "***Bayesian** approach for inference of the **C**yber-**A**dversarial **P**ower **S**ystem" (**Bayes-CAPS**)* is presented that leverages the BN formalism for constructing and inferring the Bayesian attack graphs (BAGs) in power systems. Performing Bayesian inference on these graphs enables

Bayes-CAPS to compute the likelihood that a host in the network is compromised or not given a set of alerts [11] and maps exactly to the detailed power system cyber-physical topology. The major contributions are as follows:

1. The OT threat scenario is considered in high fidelity, which differs from IT, with OT and IT prioritizing low latency and throughput, respectively. Similarly, peer-to-peer communication is rarely observed in OT but supported in the Internet. An IEC 61850 *Type 1A/P1* message for fault isolation and protection has a delay constraint of 3 ms, while for a less time-critical *Type 3* message, it is 500 ms [12]. Hence, generation of Bayesian attack graphs in a cyber-physical power system is a mixture of different types of networks and a contribution of this work.
2. The five case studies used to generate BAGs in Bayes-CAPS are constructed based on detailed realistic data flow models that capture the NERC-CIP-005 standards on electronic security perimeters for bulk energy systems.
3. From the communication networks, Bayes-CAPS contributes a method to construct BAGs using the power system threat and data-flow models [13].
4. Bayes-CAPS assesses how different evidence sources impact inference. Results with respect to varying threat strength, and on the basis of scale, accuracy, computation time, dependence on evidence, and loops, are detailed for five power system use cases. This is crucial to the accuracy of Partially Observable Markov Decision Processes (POMDP) for estimating cyber-physical state in power systems, since the Bayes-CAPS inference would update its belief state. The POMDP's belief state is crucial to reinforcement learning (RL) and hence its potential to be safely used in power system critical infrastructures.
5. The work integrates the Bayesian inference and structure learning framework into the Resilient Energy Systems Lab (RESLab) testbed as a plugin named Bayes-CAPS.

The paper proceeds as follows. Section 2 positions Bayes-CAPS with respect to related work. The Bayes-CAPS framework is presented in Section 3. Section 4 presents the BAG generation step. Section 5 details the five cyber-physical power system use cases. Section 6 presents the inference step and impact analysis of evidence. Section 7 provides the Bayes-CAPS pseudocode. Section 8 evaluates Bayes-CAPS, and Section 9 concludes the paper.

2 Power System Security Inference Background

A significant amount of background is relevant, as this work stems from several distinct areas; detailed coverage is beyond the scope of this paper. Hence, the focus in this section is on background aspects that are most relevant and critical to positioning this work, categorized into key aspects that enabled us to attain this new solution and can enable others to understand, repeat, and extend this work.

2.1 The Problem of Power System Critical Infrastructure Cyber-Physical Security Inference

A power system as a graph has cyber, physical, and cyber-physical interconnected nodes. Intrusions may propagate within the cyber portion of the network, to sequentially exploit vulnerabilities and escalate privileges over cyber nodes, finally targeting the interconnecting nodes, to affect physical devices such as breakers. The propagation is uncertain from both intruder and defender perspectives. An intruder can follow any strategy depending on its successful tactics, while a defender can only observe symptoms such as resulting network latency or contingencies.

This motivates Bayes-CAPS to use PGMs. A specific PGM variant called a Bayesian Network (BN) is considered for modeling threats using inference on attack graphs. Conventionally, for intrusion analysis, a dedicated Security Information and Event Management (SIEM) is deployed to combine and show security information, and it is isolated from an Energy Management System (EMS). By comparison, a cyber-physical EMS is a new type of EMS currently being developed by researchers with industry partners [14] which includes an early-stage attack detection and response capability based on cyber and power information. For the research and development of this capability, a cyber-physical large-scale power system testbed, RESLab, has been built and continues to be expanded to simulate complex threat and defense scenarios, build theoretical models [15], infer intrusions, and respond optimally. RESLab supports generation and emulation of power models and their communication networks [16] and allows ICS traffic such as MODBUS, DNP3, etc., to flow realistically through the network. In multi-stage cyber intrusions, there is a causal relationship between the events in the stages [17]. Hence, Bayes-CAPS develops and adopts BAGs in the cyber-physical power system setting to perform causal reasoning between each step in access paths toward an adversary's physical target.

2.2 BNs in Security

BNs are well-known for modeling decision problems under uncertainty. In network security, BNs have been proposed for over a decade. The first work on modeling attack graphs through BNs was published in 2008 [18]. BNs have also been suggested in intrusion detection systems (IDSs), such as [19] that builds an adaptive IDS for attack signature recognition and [20] that proposes an intrusion intention recognition system. Despite work on BNs for network security, recognizing adversary intention and strategy has remained challenging due to numerous uncertainties in an IT network. By contrast, OT power networks have more static architectures and prior knowledge can be applied to lessen the uncertainty and make BNs a useful tool.

2.3 BNs in Power Systems

In power systems, a main use of BNs is to perform model-based diagnostics. In [21], authors propose a probabilistic-based approach, where the electric power system is represented as a BN in the testbed at the NASA Ames Research Center. The idea of an inference-based expert system is proposed in [22] to support control center decision making under emergency. Similarly, a recent work on equipment failure estimation of power distribution system equipment, such as on-load tap changers and switched capacitors, using Bayesian inference, is presented in [23].

The use of PGMs for combined cyber-physical security analysis is less explored. Authors in [19] provide a framework for an adaptive IDS that uses BNs, and [24] performs an exact inferencing algorithm using junction tree and belief propagation. An anomaly reasoning engine is proposed in [25] that utilizes Bayesian inference on causal polytrees to produce a high-level view of the security state in a SCADA network. The work in Bayes-CAPS is motivated by [24] and [25], to propose Bayesian inference algorithms for networks of realistic size and complexity, and to recommend viable inference solutions for these networks using dynamic evidences.

2.4 Automatic Attack Graph Generation

Numerous works address automatic attack graph generation. Researchers in [26] propose novel approaches for generating attack privilege fields as prerequisites and post-conditions based on the Common Vulnerability Scoring System (CVSS) and the National Vulnerability Database (NVD). The NVD and CVSS are widely used and detailed in [27]. Authors in [28] propose an approach to construct knowledge graphs from structured data, while [29] proposes an algorithm that optimizes the network topology before generating the graph. Instead of using CVSS scores to approximate attack characteristics, our work constructs the initial BAG parameters based on the CVSS score distribution.

2.5 Bayesian Inference Algorithms

Bayesian inference is used to calculate the posterior probability of query variables, given a set of evidences. In [30], authors propose an improved likelihood weighting algorithm to extrapolate network security states using attack graphs and intrusion evidence. Authors in [18] propose to model probability metrics based on a special BAG using conditional probabilities to model interdependencies between vulnerabilities. A graphical inference engine for multiple-intrusion detection is proposed in [17] that performs belief propagation on an appropriately constructed weighted bipartite graph. Variable elimination is used in [31] to improve the accuracy and efficiency of a dynamic BN to infer intention and behavior in human-robot interaction. Authors in [32] propose an intrusion detection and prevention system for zero-days attacks using Bayesian structural and parametric learning. Junction tree based Bayesian inference is adopted in [33] to trace the most probable access paths from all possible atomic paths. In our work, we adopt Belief Propagation using Factor Graphs (BP_FG) [34], Variable Elimination (VE) [35], Pearl's Belief Propagation (PBP) [36], and Junction Trees (JT) [37].

2.6 *Context of Bayes-CAPS*

By comparison, this work details a Bayesian framework for defining and solving the inference problem of adversarial behavior, based on realistic power system electric and cyber model information, using cyber-physical attack graph analysis and Bayesian networks, as a way to better understand adversarial movement toward a physical target in these systems. The applicability to the real world power systems is a major benefit of Bayes-CAPS as opposed to other methods with a similar objective. Of particular note is the ability to incorporate known system information, which is a key benefit and highly advantageous to grid defense.

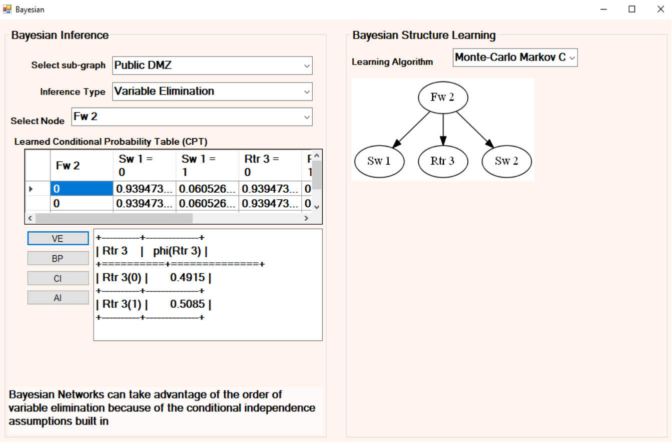
For crucial power system computation and control environments, Bayes-CAPS is versatile and allows ease of construction from cybersecurity domain experts' prior knowledge, threat models, and learning from raw data. Further, BNs are PGMs that help account for uncertainty of an adversary's behavior, which is a serious and major issue in power systems, and this work is an important contribution toward addressing that need. The approach is capable of performing the important and currently lacking function of causal reasoning between each step in the access paths of the adversary's trajectory to compromise the target.

Power system cyber-physical critical infrastructure systems are considered in high fidelity, with threats targeting OT and grid operational reliability. Critical infrastructure systems are distinct based on the essential nature of the societal functions they protect. In cyber-physical critical infrastructure, availability and integrity are of utmost importance. Integrity, availability, and timeliness are crucial; power systems must remain online. Timely delivery of accurate commands and measurements must always occur, even under extreme scenarios. Presenting and demonstrating Bayes-CAPS for five realistic power system cyber-physical case studies is a contribution of this work to advance critical infrastructure defense. Bayes-CAPS shows how BAGs can be automatically constructed and applied based on detailed realistic data flow models that real utilities will be using based on NERC CIP standards for bulk energy systems. Hence, this work will help achieve deployability and applicability in real world power systems. To further aid deployability in utilities, it is important to understand how to place and configure the monitoring tools (e.g., budget may be limited, monitoring must be prioritized based on risk and observability). Hence, the experiments and test cases in this work can also lend guidance on how the most useful evidence may be collected.

3 Bayesian Framework in CYPRES EMS

The Cyber-Physical Resilient Energy Systems (CYPRES) EMS [14] is a prototype cyber-physical EMS being developed and evaluated in RESLab [38]. The work in Bayes-CAPS is implemented in the CYPRES EMS and leverages the $pgmpy$ package [39] for inference algorithms such as Variable Elimination (VE), Belief Propagation (BP), and Infer.NET [40], as shown in the left side of the tool (Fig. 1). The inference and learning engines in CYPRES are running as python scripts. This work integrates these libraries in the CYPRES EMS to (1) provide core functionality for causal inference and (2) to provide a framework for performing inferences for various cyber-physical power system use cases.

The inference engine is at the utility control center and at the substation level in the hierarchy (see Fig. 6). In Fig. 1, the sub-graph of the utility control center network is selected, then the inference type and node are selected. The screenshot of the tool shows a sample of the analysis and results described in this paper. Specifically, the tool shows the conditional probability table (CPT) associated with the selected firewall, i.e., $Utility\ 39..Firewall\ 1368$ in the model. The scores show the conditional probability of the host being compromised if the selected firewall is compromised. The right side of the tool shows the learned structure of the BN, given raw alerts in the form of the Pandas dataframe. In the current tool, the graph-based structure learning algorithms are incorporated based on [10].



Select sub-graph: Public DMZ

Inference Type: Variable Elimination

Select Node: Fw 2

Learned Conditional Probability Table (CPT)

Fw 2	Sw 1 = 0	Sw 1 = 1	Rtr 3 = 0	F 1
0	0.939473...	0.060526...	0.939473...	0
0	0.939473...	0.060526...	0.939473...	0

```
+----------+-------------+
| Rtr 3    |   phi(Rtr 3) |
+==========+==============+
| Rtr 3(0) |       0.4915 |
+----------+-------------+
| Rtr 3(1) |       0.5085 |
+----------+-------------+
```

Bayesian Networks can take advantage of the order of variable elimination because of the conditional independence assumptions built in

Learning Algorithm: Monte-Carlo Markov C

Fig. 1: Screenshot of Bayes-CAPS tool for Bayesian inference and learning in the CYPRES EMS prototype to infer adversary behavior in a cyber-physical power system.

4 Bayesian Attack Graph Model Generation

Cyber-induced threats often initiate from the cyber layer of a cyber-physical network, as witnessed in events such as Stuxnet, Ukraine, and Colonial Pipeline attacks. Hence, for studying intruder propagation to reach its target, it is essential to model the communication network for each physical system in detail. Hence, this work first constructs the communication network for the power systems under study, following the approach for the synthetic communication network built for a 2000-bus electric grid model [16]. Further, this work interfaces the network models with the power systems and constructs a Bayesian variant of attack graphs for cyber-physical systems. In the revision, Sections 2, 5, and 9 are updated accordingly, (1) to enhance reproducibility now by clarifying details about the datasets and models available now, and (2) to explain the next steps in the direction to opensource additional code and data.

An attack graph is a graphical model that represents how the vulnerabilities in a network can be sequentially or parallelly exploited, showing different paths an adversary can take to reach its target. Attack graphs can be expressed in many forms and may be *state-based* or *logical* [24]. A node in a state-based attack graph represents the security state as the combination of compromised hosts. Scalability is an issue, as the network becomes dense with increasing host connectivity. Logical attack graphs represent dependencies between exploits and security conditions. In this work, logical attack graphs are studied.

4.1 *Logical Attack Graph Creation*

A scenario is adopted for purposes of illustration, mimicking the control center of the Iranian Natanz plant targeted by the Stuxnet worm (Fig. 2, left, adopted from Fig. 5 of [41]). The enterprise control network contains $WinCC\ Web\ Client$, $Historian\ Web\ Client$, and $PCS7\ web\ client$. WinCC is a SCADA and HMI system, and PCS7 is the Distributed Control System. These clients accessed the web server through $Web\ Navigator$, $PCS7\ Web\ Server$, and $CAS\ Server$ in the demilitiarized zone (DMZ). These servers could interact with the process control components such as $Engineering\ Workstation$, $Simatic\ Process\ Historian$, and $PCS7\ OS\ server$ to control the PLCs and field devices.

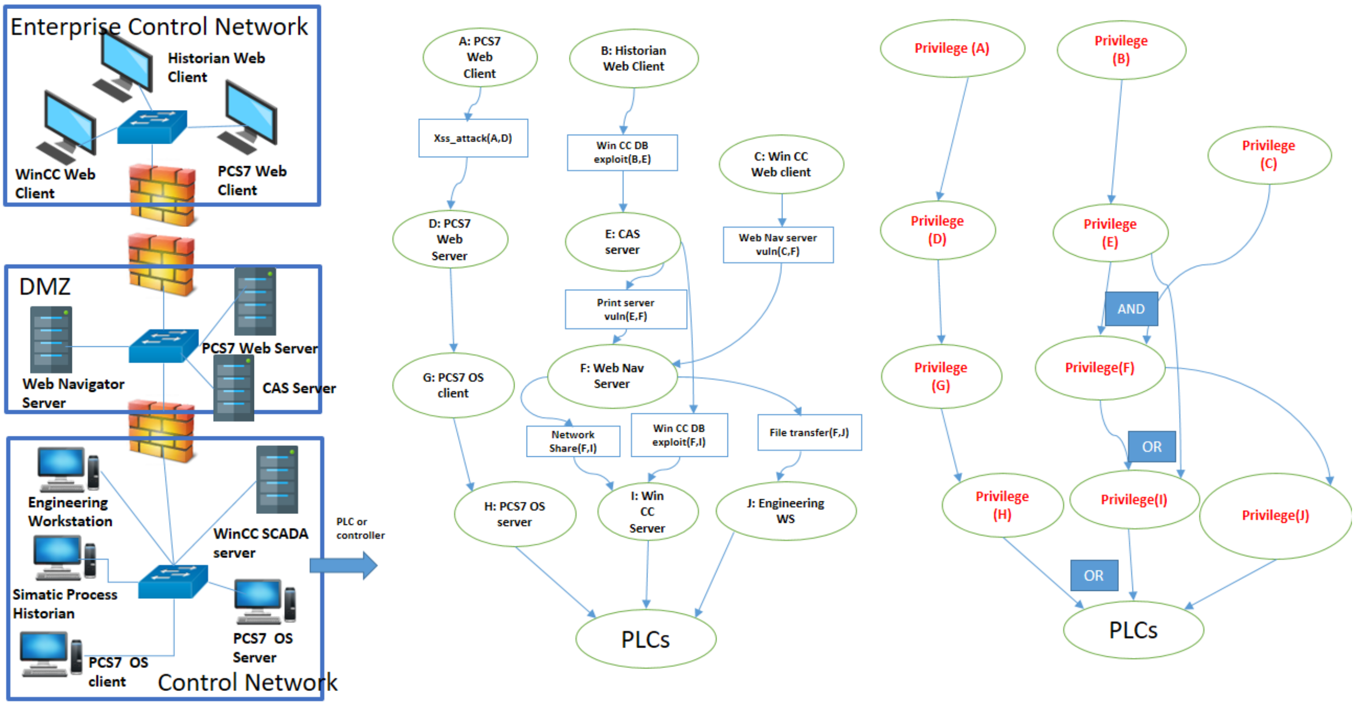


Fig. 2: Bayesian Attack Graph (BAG) modeling example: An industrial control system (ICS) supervisory control and data acquisition (SCADA) network (left) is converted to a logical attack graph model (center, Section 4.1), then to a BAG using a BN (right, Section 4.2).

To create the logical attack graph (Fig. 2, center), several inputs are used: (1) the configurations of the Access Control Lists (ACLs) in the three firewalls, (2) the vulnerabilities in the web servers in the DMZ, and (3) the potential target, e.g., root access to $Engineering\,Workstation$ or $WinCC\;SCADA\;server$. Security conditions are represented as circle nodes, e.g., adversary has privilege over the host $D : PCS7\;Web\;server$. Vulnerabilities are represented as rectangle nodes, e.g., in $Xss\_attack(A, B)$ an adversary with privilege over $A : PCS7\;Web\;client$ can then exploit the cross-site scripting vulnerability in $B : PCS7\;Web\;server$. The probability of the exploitation of the vulnerability to reach a security condition is considered to be based on the *Base Score* of the CVSS (CS) [27]. CVSS scores are in the range of 0-10. For each day in the NVD, the number of exploits and distribution across CVSS scores is calculated (Fig. 3), based on actual reported findings. Then, using this distribution, the attack graph's nodes are allocated for a vulnerability (v) with its probability of being exploited (p_v), where p_v is based on the linear relation $p_v = f(CS)$.

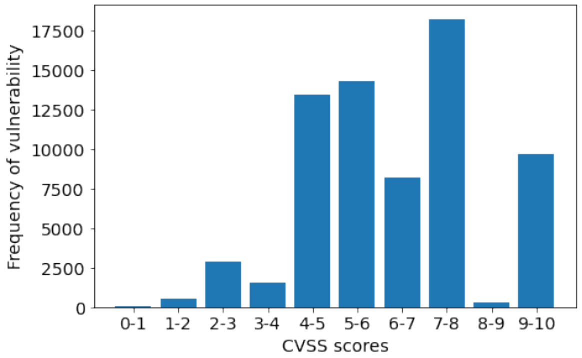


Fig. 3: Exploitation distribution of different vulnerabilities across CVSS scores existing in the NVD database [27], over the range of 0-10 on the day of inference.

4.2 Conversion to Bayesian Attack Graph

The fusion of a BN with the attack graph makes it a BAG. A BN is introduced for dynamic analysis of attack graphs in [42]. The nodes represent *random variables*, and the directed edges represent dependencies between them, forming a Directed Acyclic Graph (DAG) [24]. If monotonicity is assumed, once an adversary escalates a privilege, it never relinquishes it; then, one can remove duplicate paths to construct the DAG. The joint probability distribution of the BAG (Fig. 2, right) can be written as:

$$\begin{aligned} P(A,B,C,D,E,F,G,H,I,J,PLCs) &= P(A)P(B) \\ P(C)P(D \mid A)P(G \mid D)P(H \mid G)P(E \mid B)P(F \mid C,E) \\ P(I \mid F,E)P(J \mid F)P(PLCs \mid H,I,J) \end{aligned} \tag{1}$$

Each node in the BAG represents a security condition, e.g. $Privilege(A)$. The CPT of the BAG is then computed as the combined effect of a vulnerability in a network, as used in [43]. The local CPT of the nodes with logical AND and OR conditions in the BAG are computed as Eqn. 2, from [24]. A logical AND signifies that all the permissives are necessary to compromise a node X_i. A logical OR signifies any one permissive is sufficient to compromise a node. The immediate parents of the node X_i are $\mathbf{pa_i}$, and the probability that vulnerability v_j is exploited is p_{v_j}.

$$\begin{gathered} P\left(X_i|\mathbf{pa}_i\right) = \begin{cases} 0, & \exists X_j \in \mathbf{pa}_i | X_j = 0 \\ \prod_{j:X_j} p_{v_j}, & \text{otherwise} \end{cases} \\ P\left(X_i|\mathbf{pa}_i\right) = \begin{cases} 0, & \forall X_j \in \mathbf{pa}_i | X_j = 0 \\ 1 - \prod_{j:X_j} \left(1 - p_{v_j}\right), & \text{otherwise} \end{cases} \end{gathered} \tag{2}$$

The computational steps of the BAG model generation of BayesCAPS are detailed in Lines 1-10 of Alg. 1. Five use cases are considered in this work, described next.

5 Power System BAG Model Generation

Since power systems are critical infrastructure, the real systems have constraints on data sharing, and the real model is Critical Energy/-Electric Infrastructure Information (CEII). Hence, the term *synthetic* is adopted to refer to models based on characteristics of real systems that are of the same level of detail, scale, and complexity, but do not contain any CEII or represent any specific real-world system, and hence can be shared. For the synthetic models below, Bayes-CAPS generates BAGs, based on cyber-physical architecture and components as detailed in [16], and performs studies that utilize the models to understand how well the inference techniques work in this environment.

5.1 Threat Model

The threat model in the BAGs that we generate captures the type of vulnerability exploited, as well as the set of permissives that need to be satisfied to exploit the vulnerability in the succeeding node (child node). Based on the type of vulnerability, the CVSS score implicitly reflects the attack strength based on its exploitability component. The set of permissives is based on the parameter p_{OR}, the probability of incorporating OR logic (as opposed to AND logic) in the step(s) between the parent and child nodes. Hence, the threat model for adversary movement is logical in nature, considering OR and AND gates for vulnerability exploitation, with threat intensity regulated through p_{OR}. The notion can be explained through the Darkside Ransomware-as-a-Service attack on the Colonial Pipeline, where three different parents or affiliates were involved to plant the Backdoor Smokedham in the victim's environment. The UNC2659 affiliate first exploited the CVE-2021-20016 vulnerability of a SonicWall firewall to install Teamviewer and get remote access, while the affiliate UNC2628 utilized the Beacon command and control Botnet framework to run the Mimikatz, a credential theft tool for privilege escalation. Finally, the affiliate UNC2465 plants the .NET backdoor, Smokedham, by sending a phishing email. Hence, for this incident, the event of backdoor access had a $p_{OR} = 0$ as all affiliates needed to execute their steps to finally accomplish the goal. The higher the value of p_{OR}, the higher the chance of exploitability, because there would be more successful ways to target the victim. This method of modeling the threat makes the threat framework generic to attacks of diverse intensity and exploitability.

Other threat scenarios of a false data and command injection attack, using an Address Resolution Protocol (ARP) spoof based Man-in-The-Middle (MiTM) attack in the RESLab testbed [44], as well as communication loss via Denial of Service (DoS) [38], are considered for learning the BAG based on the sensor data from different locations in the emulated network. The specific targets in these attacks are branch statuses and generator set points, based on multiple element contingency selection in large-scale power systems [45]. For extracting real-time data during the threat emulation, the fusion engine [46] is used. The scenario's dataset is available at IEEE Dataport [47]. Bayesian structure learning is considered to understand the dynamics of the attack by modeling the nodes of the BAG as the cyber and physical features extracted at different location of the network. The dynamic details of the attack are elaborated in Fig. 2 of [44], including a detailed timing diagram of the ARP cache poisoning attack on the substation router and the field devices. This forms the basis for utilizing the IP and MAC address based features from four different locations in the network. The learning techniques are adopted from [10]. Interested readers are encouraged to review the prior works for an in-depth grasp of this threat model, since the details are outside the scope of this paper.

5.2 Case Studies

Using the steps described in Sect. 4 and Alg. 1, the BAGs are generated for different communication networks based on a hierarchical network [16], illustrated in Fig. 6. In these networks, an electric grid is divided among several utilities, where each utility owns a set of substations. The utilities or the market participants interact with the balancing authorities at the highest level of the hierarchy. Modeled components include firewalls, routers, RTACs, switches, DMZs, and relays. The cases are detailed in [15, 48] and available [49].

The construction of the graphs is automatic, with the test system generation based on realistic power systems [16, 48] as the first step. An Attack Graph Template (AGT) is used to construct the attack graph based on inputs that would be collected from the real world power systems, detailed in [50, 51]: the firewall rules and information about ports and services running to translate into CVSS scores. Those data can be collected using existing available tools. From those inputs, a cyber-physical attack graph is built (detailed in [50], code for CYPSA-Live [52]) that gives the access paths that can be exploited to reach a high-impact target. The process is documented in detail in [16, 48], with data in [49]. The attack graph construction and subsequent physics-based risk analysis enables a key value proposition for many different cyber-physical analyses, including this one, that moves from static attack graph to dynamic attack graph in aim of continually mapping a dynamic adversary.

When the attack graph is constructed, it is based on the connectivity model and refined based on the known vulnerability information to preclude links that are unlikely based on the lack of available vulnerability information. To address zero-day vulnerabilities, it is important to note that data-driven probabilistic models depend on the symptoms of an intrusion. Assuming the probability a given node X is compromised is $P(x)$, even if a vulnerability is missed while constructing the BAG (e.g., zero-day exploit), the symptoms of the intrusion would enforce the update of the posterior probability of $P(x)_{post}$. However, symptoms may not be reliable due to untrustworthy sensors. Hence, other works, e.g., [7], that address aleatory uncertainty through the notion of ignorance (analogous to the impact of a zero-day exploit on the symptoms of an intrusion), can be considered. Further, alerts generated from IDS can act as a data source for the structure learning problem to learn the structure of an attack graph based on the prior structure provided by experts [10].

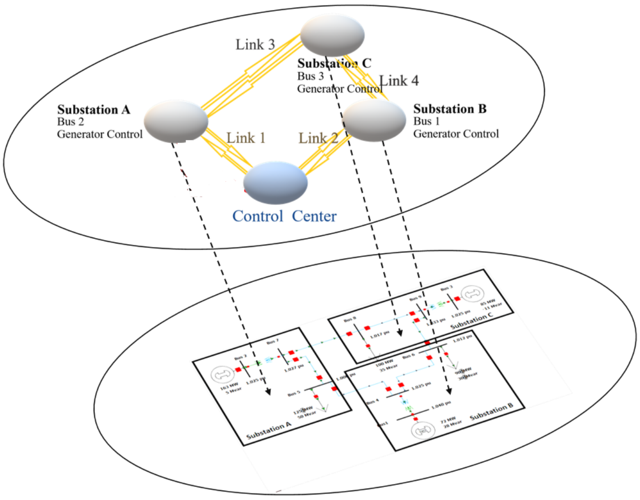


Fig. 4: WSCC 9-bus cyber-physical model with control center and substations, figure adopted from [53].

5.2.1 Single Substation Model

A utility control center with a single substation (Fig. 6) provides a simple benchmark test case for evaluating effects of different parameters on the performance of Bayes-CAPS. The single substation model also serves as a module in the code that is able to be replicated and parameterized in larger models. This model forms a BN with 29 nodes (17 in the UCC, 12 in the substation). Fig. 5 shows the BAG.

5.2.2 WSCC Case with 3 Substations

A three-substation network is created based on the WSCC 9-bus case [54], which consists of 4 broadcast domains, one each for the substation and one for the main control center (Fig. 4). This forms a BN with 47 nodes (17 in the UCC, 30 in 3 substations).

5.2.3 CyPSA 8-substation Model

The cyber-physical situational awareness (CyPSA) 8-substation test case [48] contains a node-breaker topology with detailed cyber and physical interconnections at the substation level. Each substation has multiple buses and

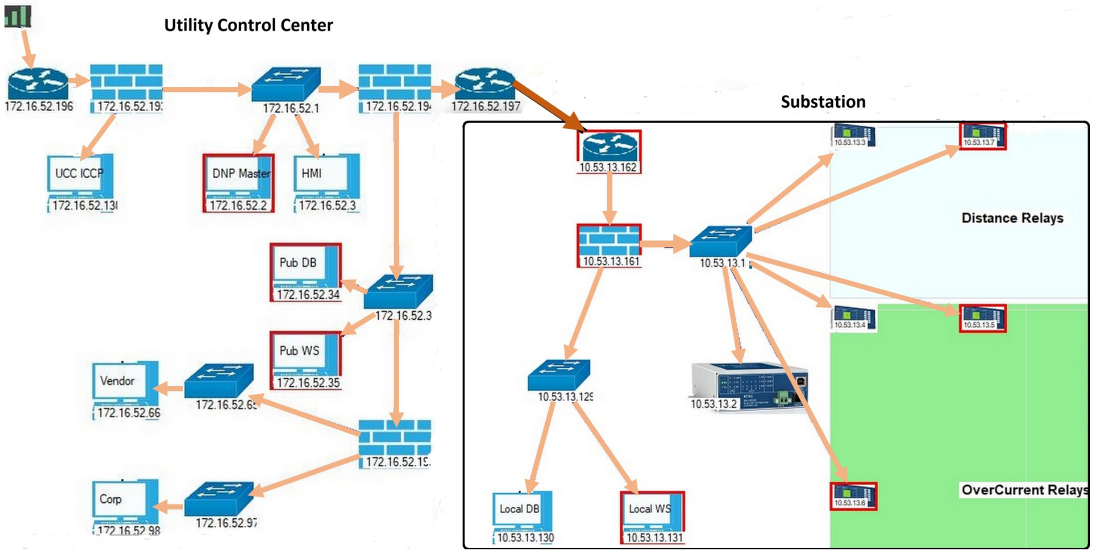


Fig. 5: Bayesian attack graph for the single substation model; nodes circled in red indicate evidence of compromise.

control devices, which makes the model dense. There are 52 electrical nodes. Relays, breakers, firewalls, and routers are modeled, detailed in [48]. This forms a BN with 98 nodes (17 in the UCC, 81 in 8 substations).

5.2.4 IEEE 300-bus System The IEEE 300-bus power system case is made cyber-physical as detailed in [55]. This case is used to study the scalability of the proposed model, as it consists of 4500 IP-addressable devices, with 1301 operational devices, i.e., relays, and 2384 non-operational devices, e.g., fault recorders, alarm systems, and batteries. The model has 300 substations, grouped into 20 areas for the experiments. Hence, a total of 21 broadcast domains are considered, and the BN has 217 nodes. This grouping reduces model complexity by reducing the number of nodes from 4500 to 217.

5.2.5 2000-bus Synthetic Grid The cyber-physical synthetic electric grid test case [16, 56] with 1250 substations, 2000 buses, 3206 branches, and 544 generators is considered in emulation in RESLab under threat scenarios that cause multiple element contingencies. The model is built based on public information and statistical analysis of real power systems, without disclosing any real system information. The emulation of this system is detailed in [38].

5.3 Discussion

This work is the first attempt on building BAGs for a large-scale power system model. For BAG nodes that represent the compromise of a relay or RTU, it causes a contingency in the physical (electrical) domain, modeled in PowerWorld Dynamic Studio (PWDS). In the present work, the power domain impact is modeled in a deterministic manner, dictated by the circuit behavior. For instance, outage of a generator may cause a transmission line to overload and trigger an over-current relay, say $R1$, to open a circuit breaker, which changes the power flow based on the circuit's characteristics, and may trigger some other protection device, say $R2$. In this scenario, the conditional probability of $P(R2|R1)$ is 1, assuming $R1$ to be the parent and $R2$ a child in a sub-graph of the BAG. Integration of renewable generation and variable load profiles can be considered in future work to model probabilistic behavior of multi-stage contingencies in the physical domain.

A major goal driving our work is to provide a decision support capability under the loss of visibility caused by cyber threats. Conventionally, the control problems in power systems have been considered and addressed using optimization theory. Instead, our work considers how to apply known model information while leveraging RL to improve both the timeliness and accuracy of stakeholder decision making. Using RESLab developed for this purpose, it supports formulating and validating the problem as a POMDP. Using Bayesian inference, we compute the posterior probabilities which act as the belief state within the POMDP. In POMDP, once the belief states are learned, the agent is agnostic to how the posterior probabilities were computed. The solution for the POMDP makes use of BRL which is based on Bayesian inference. Hence, in the current work, we synthetically update the inference against the BAG by altering the p_{OR} values. The environment in the RL space would encounter variability from updated inferences with every evidence. Hence, we analyze the impact of evidence on the inference.

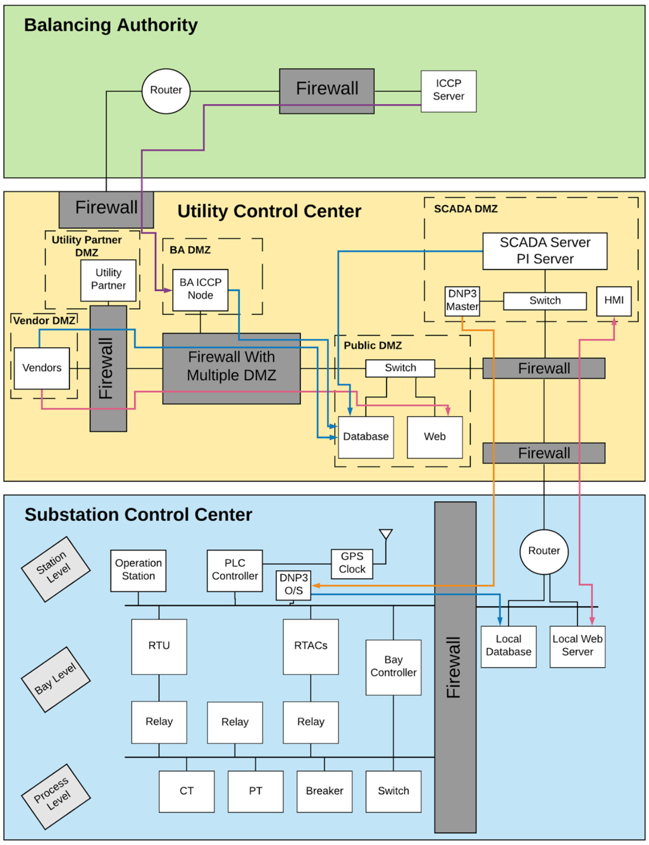


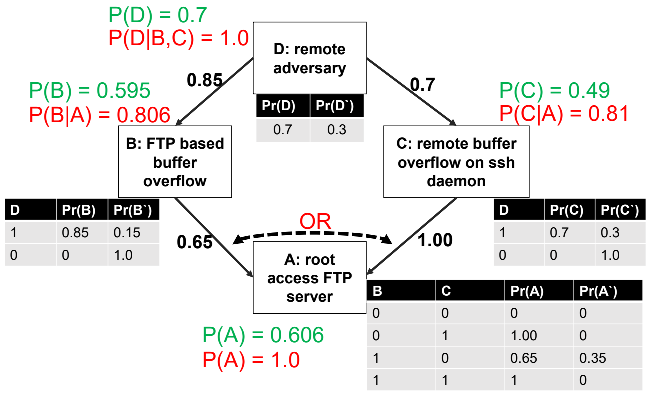
Fig. 6: Hierarchical architecture of the communication network of Power System with the links indicating the IT and ICS traffic.

It is important to help stakeholders understand how different evidences impact the inference, which impacts the ability of RL to provide decision support. Analyzing the impact on decision support thoroughly involves the fundamentals of RL and is outside this paper's scope. However, as an example, consider an RL episode, where an intruder goal's is to compromise a PLC controller, as in Fig. 2. In the episode, the adversary will be compromising the $CAS\ server$, $PCS7\ OS\ client$, $Win\ CC\ Server$, etc., sequentially. Similarly, the defender engine will have to take certain steps to prevent the intrusion. At every stage, the IDS within the RL environment finds an evidence, and the belief state is updated based on Bayesian inference, which impacts the decision/policy of the defender.

This research with power system BAGs reveals a challenge, as it would benefit from real adversary behavioral data to compare. Modeling the prior probabilities of the BAGs needs emulation of red team activities over a long-term duration to capture how an intruder thinks and executes. Estimating the prior probabilities with a large-scale dataset would require exposing the RESLab testbed to an outside world for threat integration which is currently out of scope due to constraints on allowing outside researchers to deploy their threat strategies. For simplicity, we have hence limited the usage of the data from the emulation to a small network within RESLab for learning the structure of the BN.

6 Approach for Inference on the BAGs

The BAGs generated based on the cyber-physical model have static attributes if the events within the system do not update the properties like the posterior conditional properties. Each event of intrusion detection must update the BAG attribute by performing inference, called Bayesian inference. It is used to calculate the posterior probability of query variables, given a set of evidences (Fig. 7). It computes the unconditional probability distributions to determine the probability that an adversary can reach a security condition. The inference algorithm performs marginalization, i.e., summing out the probability of a random variable given the joint probability distribution with other variables. In this work, we adopt and compare four different inference algorithms.



Pr(D)	Pr(D`)
0.7	0.3

D	Pr(B)	Pr(B`)
1	0.85	0.15
0	0	1.0

D	Pr(C)	Pr(C`)
1	0.7	0.3
0	0	1.0

B	C	Pr(A)	Pr(A`)
0	0	0	0
0	1	1.00	0
1	0	0.65	0.35
1	1	1	0

Fig. 7: Evaluation of the impact of evidence in a small a BAG, showing CPTs for each node. Initial prior probabilities (e.g., the prior on D is $P(D) = 0.7$) are shown in green. Posteriors (given *evidence* that A is compromised) are shown in red.

6.1 Understanding the Notion of Evidence

To illustrate the notion of evidence, we consider a simple BAG from the paper [57], shown in Fig. 7. Here, the remote adversary targets root access to an ftp server by performing buffer overflow attacks by means of two paths, through B (exploiting an ftp vulnerability) or C (exploiting a ssh vulnerability). The figures shown in the edges of the BAG are the CVSS scores. The unconditional probability for the given BAG, given the prior probability of $P(D) = 0.7$, is computed as:

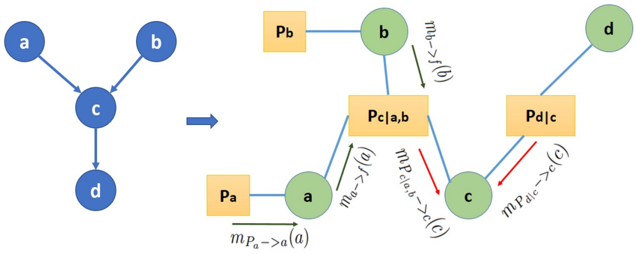


Fig. 8: The factor graph with message passing for the BAG in Fig. 7.

$$\begin{aligned} P(B) &= \sum_{D} P(B|D)P(D) = 0.595 \\ P(C) &= \sum_{D} P(C|D)P(D) = 0.49 \\ P(A) &= \sum_{B,C,D} P(A,B,C,D) = 0.606 \end{aligned} \tag{3}$$

Then, assume an evidence is obtained at the node A that confirms it to be compromised, i.e., $P(A) = 1$. Then, the posterior probabilities at node C, B, and D are altered based on Bayes theorem:

$$\begin{aligned} P(C|A) &= P(A|C)P(C)/P(A) = 0.49/0.606 = 0.81 \\ P(B|A) &= P(A|B)P(B)/P(A) = 0.806, \text{ as} \\ P(A|B) &= \sum_{C} P(A|B=1,C)P(C) = 0.8215 \\ P(D|B,C) &= P(B,C|D)P(D)/P(B,C) = 1.0, \text{ as} \\ P(B,C|D) &= P(B|D)P(C|D) = 0.85 \times 0.7 = 0.595 \\ P(B,C) &= \sum_{A,D} P(A,B,C,D) = 0.4265 \end{aligned} \tag{4}$$

Every time a new evidence is obtained, the BAG is dynamically updated, as in Fig. 7. Updating the posterior based on multiple evidences is similar to our prior work on multi-sensor fusion work [7] based on DSTE and its rules of combination.

6.2 Belief Propagation with Factor Graphs (BP_FG)

A factor graph is a bipartite graph containing variable nodes and factor nodes, and the edges always connect nodes of different types. The joint probability distribution for the BAG in Fig. 7, is the following,

$$P(a,b,c,d) = P_a(a)P_b(b)P_{c|a,b}(c|a,b)P_{d|c}(d|c) \tag{5}$$

whose factor graph is shown in Fig. 8, where green and yellow denote the variable and factor nodes, respectively. We then compute the marginals and conditionals by passing messages (or propagating beliefs) on the factor graph. This algorithm is widely known as the sum-product algorithm, based on three steps:

1. Marginalization at variable node: Marginals are the product of all incoming messages from neighbour factors (Eq. 6),

$$P(v) = \prod_{f \in F_v} m_{f \to v}(v) \tag{6}$$

where the $m_{f \to v}(v)$ is the message from neighboring factor node f to variable node v, and F_v is the set of all factor nodes neighbor to v. Then, for example, $P(c)$ in Fig. 8 is the following:

$$P(c) = m_{P_{c|a,b}->c}(c)m_{P_{d|c}->c}(c) \tag{7}$$

2. Operation at factor node: Messages from factors sum out all variables except the receiving one. For example, the message from factor node $P_{c|a,b}$, say f, to variable node c in Fig. 8 is the following:

$$m_{f->c}(c) = \sum_a \sum_b f(c|a,b) m_{a->f}(a) m_{b->f}(b) \tag{8}$$

3. Operation at intermediate variable node: Messages from variables are the product of all incoming messages except from the receiving factor. For example, the message from variable node a to factor node $P_{c|a,b}$ in Fig. 8 is the following:

$$m_{a->f}(a) = m_{P_a->a}(a) \tag{9}$$

6.3 Pearl's Belief Propagation (PBP)

Pearl's belief propagation (PBP) computes the belief distribution of a random variable X in the BN. The distributions are computed using three types of parameters: a) causal support (π), b) diagnostic support (λ), and c) the CPT. The CPT is described in [25] and illustrated in Fig. 7. The algorithm initiates from the node where a new evidence is obtained, following three steps [25]:

1. Belief updating: The node updates its belief based on the message it receives from parents, through causal support (π), and the message it receives from children, through diagnostic support (λ):

$$Bel(X) = \alpha\lambda(X)\pi(X) \tag{10}$$

2. Bottom-up propagation: The node computes a new message $\lambda_X(u)$ based on its CPT and messages λ received from its children.

$$\lambda_X(u) = \sum_X \lambda(X) P(X|u) \tag{11}$$

3. Top-down propagation: The node computes a new message π and that it sends to its children. The new message $\pi_{Y_j}(x)$ for its j^{th} child Y_j is calculated as follows:

$$\pi_{Y_j}(X) = \alpha\pi(X) \prod_{k \neq j} \lambda_{Y_k}(X) \tag{12}$$

Two different variants are considered, $PBP-P$ and $PBP-T$, where $PBP-P$ parallely starts the belief propagation and can update the posteriors for BAGs with loops. $PBP-T$ sequentially performs belief propagation and only works in BAGs without loops.

6.4 Variable Elimination (VE)

The Variable Elimination (VE) method groups together factors that involve the same variables, then marginalizes those variables. For example, considering the BAG from Fig. 2, the probability that an adversary gains to access the PLCs is $p(PLCs)$,

$$\begin{aligned} P(PLCs) = & \sum_{A,B,C,D,E,F,G,H,I,J} P(A)P(B)P(C) \\ & P(G \mid D)P(D \mid A)P(H \mid G)P(E \mid B)P(F \mid C,E) \\ & P(I \mid F,E)P(J \mid F)P(PLCs \mid H,I,J) \end{aligned} \tag{13}$$

where after factorization, i.e., splitting the joint distribution into conditional and marginal probabilities, Eq. 13 above becomes:

$$\begin{aligned} P(PLCs) = & \sum_{H,I,J} P(PLCs \mid H,I,J) \sum_{F,E,G} P(H \mid G) \\ & P(J \mid F)P(I \mid F,E) \sum_C P(C)P(F \mid C,E) \\ & \sum_B P(B)P(E \mid B) \sum_D P(G \mid D) \sum_A P(A)P(D \mid A) \end{aligned} \tag{14}$$

Then, Eqn. 14 is evaluated from right to left by recursively eliminating all the variables in the BAG except the $PLCs$ by following the elimination order $=A,D,B,C,F,E,G,H,I,J$ using the bucket elimination algorithm [35].The complexity of the bucket elimination algorithm is $O(nw*)$, where n is the number of nodes, and $w*$ is the induced tree width [35]. The induced tree width is dependent on the BAG structure.

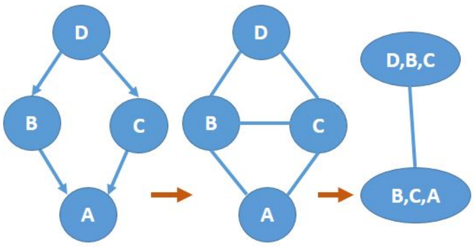


Fig. 9: The junction tree formed from the BAG, using moralization followed by triangulization to construct the chordal graph, for the example in Fig. 7.

The VE method is advisable for BAGs with more depth and less induced tree width, i.e., when number of vulnerabilities per node is low or access paths are long. For large systems, exponential blow-up while computing marginal probabilities is addressed by identifying factors in the joint distribution that depend on selected variables, then computing them once and storing the results [24]. Finding optimal elimination order is dependent on the induced graph (a graph generated at every stage of variable elimination). The optimal order depends on the BAG topology and node elimination order criteria such as min-neighbours, min-fill [24].

6.5 Junction Tree (JT)

The Junction Tree (JT) method uses the message passing algorithm as in [34] for loopy BNs. The objective is to create a tree where each node represents a collection of random variables (security states in BAGs) and apply the message passing scheme to compute the unconditional probabilities. We use the *chordal graph* method for obtaining the JT in this work, as shown in Fig. 9 . A chordal graph is a graph in which every cycle of length four and greater has a cycle chord. Moralization converts the BAG to an undirected graph, triangulazation converts the undirected graph to a chordal graph, and the chordal graph is converted to the JT.

Assuming the BAG has discrete nodes with binary values (compromised or not), JTs scale in time and space as $O(|F|r^s)$, where $|F|$ is the number of factors, $r = 2$ possible discrete values, and s is the size of the largest factor [24] which depends on the network topology. Higher interconnectivity correlates with a larger s value.

7 BAG Inferencing Pseudocode

The following variables and parameters are defined for the experiments. N is the total number of nodes, u is the maximum number of parents per node, C is the number of clusters, N_Per_Ctr is the number of nodes per cluster, Sim_Count is the number of simulation for a unique configuration, dag is the adjacency matrix representing the graph, etc.

Alg. 1 provides the overall Bayes-CAPS pseudocode. First, it constructs the DAG ($ConstructDAG()$, Alg. 2). Then, it converts the DAG to a BAG by building the conditional probability table for each node ($CreateCPD()$, Alg. 3). For the use cases, Bayes-CAPS constructs the initial BAGs. Then, Alg. 1 infers the posterior probabilities using four different techniques (Section 6). The inference algorithms are tested by changing the `engine` variable. Bayes-CAPS uses `var_elim_inf_engine` for variable elimination, `pearl_inf_engine` for Pearl's belief propagation, and `jtree_inf_engine` for junction tree based inference. The functions `mk_net`,`bnet_to_fgraph`, `belprop_fg_inf_engine`, `enter_evidence`, and `marginal_nodes` are utilized from `BayesNet` libraries.

Alg. 2 and Alg. 3 are used in Alg. 1 as follows. $ConstructDAG()$ (Alg. 2) creates initial DAGs, without assuming knowledge of the network ahead of time, by randomly picking edges while meeting the constraint of u. The variable dag_mod is the modified DAG

Algorithm 1 Bayes-CAPS Pseudocode

```
 1: Define $C$, $N$, $Sim\_Count$, $dag$, $p_{OR}$
 2: for $u$ = 2 to 5 do
 3:     for $e$ = 0 to 3 do
 4:         for $sim$ = 1 to $Sim\_Count$ do
 5:             $dag = \textbf{ConstructDAG}(u, C, N, dag)$
 6:             $bnet = mk\_net(dag, C * N, node\_type)$
 7:             for $i$ = 1 to $C * N$ do
 8:                 $d = dag(:, i)$
 9:                 $bnet.cpd(i) = \textbf{CreateCPD}(p_{OR}, d, i, bnet)$
10:             end for
11:             $fg = bnet\_to\_fgraph(bnet)$
12:             $eng = belprop\_fg\_inf\_engine(fg)$
13:             $evi$ = vector with 1 to $e$ set 1 rest 0
14:             $upd\_eng = enter\_evidence(eng, evi)$
15:             define $posterior$ to store updated probabilities
16:             for $j$ = 1 to $C * N$ do
17:                 $marg = marginal\_nodes(upd\_eng, j)$
18:                 $posterior(j) = marg$
19:             end for
20:         end for
21:         Compute average computation time
22:     end for
23: end for
```

Algorithm 2 $ConstructDAG(u, C, N, dag)$

```
 1: for $j$ = 1 to $C$ do
 2:     Define $dag\_mod$ for cluster $j$
 3:     for $i$ = 2 to $N$ do
 4:         Create edges randomly for each node $i$ in cluster $j$ having maximum number of parents $u$
 5:         Update $dag\_mod$
 6:     end for
 7:     Update $dag$ for cluster $j$ from $dag\_mod$
 8:     Create random edges between any node in the cluster $j$ and the rest of the clusters.
 9:     Update $dag$ with new edges.
10: end for
11: return $dag$
```

used to update the dag for each cluster; dag is then used to create links between the clusters. For the cyber-physical power system models, the DAGs are constructed based on the topology generated from the hierarchical synthetic communication models (Section 5), instead of Alg. 2. To convert the DAGs to BAGs, $CreateCPD()$ (Alg. 3) constructs the conditional probability distribution of each node. The npa computes the number of parents of node i. Parentless nodes are assigned the prior probabilities based on the CVSS scores, CS, discussed in Section 4.2. The nodes with parents are assigned the conditional probabilities depending on the probability of logical OR, p_{OR}, and Eqn. 2 (Line 6-10 of Alg. 3).

In the experiments, N (Alg. 1, line 1) and node density dictated by u (Alg. 1, line 2), are varied. In every experiment, 20 simulations (Alg. 1, line 1, Sim_Count) are run for calculating the average computation time and average accuracy. The u is varied from 2 to 5 (Alg. 1, line 2), since most attack-originating nodes exhibit a fan-in in that range (Fig 7 of [58]). The complexity and stealthiness of attack depends on both u and p_{OR}. The dynamic update of the posterior probabilities (Alg. 1, lines 13-19) based on the alerts will help update access paths to critical assets such as PLCs, relays, etc.

8 Results and Analysis

In this section, the experimental results are presented that compare the performance and accuracy of the Bayes-CAPS algorithms. The results in this section leverage the BayesNet MATLAB library [59].

Algorithm 3 $CreateCPD(p_{OR}, dag, i, bnet)$

```
 1: $npa = sum(dag(:, i))$
 2: if $npa = 0$ then
 3:     $CS = getCVSS(npa)$
 4:     $cpt = f(CS)$        ▷ Assign prior probabilities (Section 4.2)
 5: else
 6:     if $rand(1) \leqslant p_{OR}$ then
 7:         $cpt = OR\_CPT(prob)$
 8:     else
 9:         $cpt = AND\_CPT(prob)$
10:     end if
11: end if
12: return $tabular\_CPD(bnet, i, cpt)$
```

8.1 Analysis of Inferencing Algorithms on Random BAGs

Before considering the BAGs generated for the cyber-physical power system case-studies, we first consider random construction of BAGs, to study the sensitivity of inference, with respect to graph density and impact of evidence. Since more vulnerabilities make the BAG dense, evidence of intrusion detection at a node is useful, because it assists in updating the posterior belief on a node. The IDSs may not always detect a specific exploitation, e.g., due to disabled pre-processors to prevent latency. Moreover, the nodes monitored for security may vary, effecting the evidence. Hence, in the experiments, it is studied whether the evidence can reduce the computation time and improve accuracy of inference.

The substation network in Fig. 5 has 6 cyber nodes and 6 physical devices, e.g., relays. In the experiments, N is varied from 6 to 14. Fig. 10(a) and Fig. 10(b) evaluate average computation time without and with evidence, respectively, using BP_FG. Fig. 10(a) shows that as N increases and as u increases, the average computation time increases, e.g., 0.2 s (u=2) to 0.27 s (u=5) in the 14-node BAG. The probable reason is that an increase in u causes more multiplicative operations in the factor node (step 3 of BP_FG, Section 6.2). From Fig. 10(b), the evidence reduces computation time, e.g., by $\sim$ 50% in the 6-node BAG.

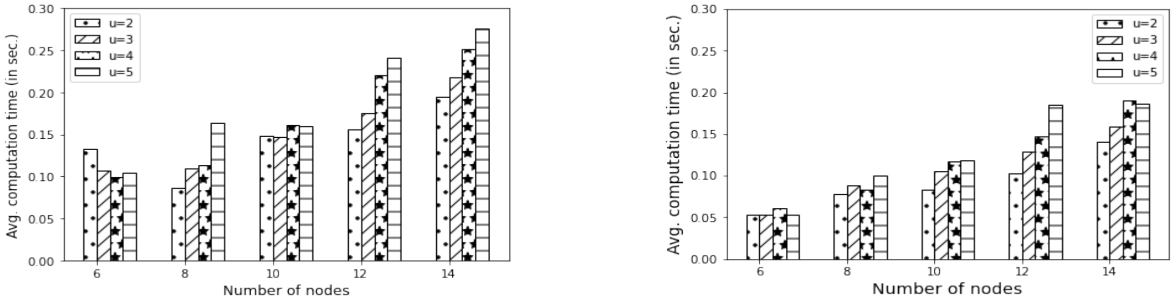
Fig. 11(a) and Fig. 11(b) evaluate computation times using VE. Fig. 11(a) shows increased computation time with increased number of nodes. Unlike in BP_FG, varying u did not effect the computation time in a specific pattern. For a fixed number of nodes, as u increases, the induced width of the tree must increase with a decrease in the depth of the tree. Hence, we would expect the computation time to grow as u increases, but the complexity also depends on the order of elimination which affects the size of the intermediate factor. The evidence reduces the computation time of VE. Fig. 11(b) shows the reduction from almost 0.06 s to 0.02 s when a single evidence was found (for N = 6, u=2).

Fig. 12(a) and Fig. 12(b) evaluate computation times using PBP. For larger N, it is observed that as u increases, the computation time also increases from 0.1 s (u=2) to 0.16 s (u=5) for the 14-node BAG. The increase in computation time with u is due to increase in number of *top-down propagation* operations that involve multiplicative operations, as discussed in Section 6.3). In Fig. 12(b), the computation time reduces from almost 0.07 sec to 0.02 sec when a single evidence is found (for N= 6, u=2).

The JT method is preferred because it works when there are loops in the BAGs. Fig. 13(a) and Fig. 13(b) evaluate the JT method without and with evidence. Unlike the previous three methods, the computation time does not increase as N increases, because the number of nodes in the JT does not dependent completely on the number of nodes in the original graph. The structure and size of the JT is dependent on the chordal graph, formed as discussed in Section 6.5.

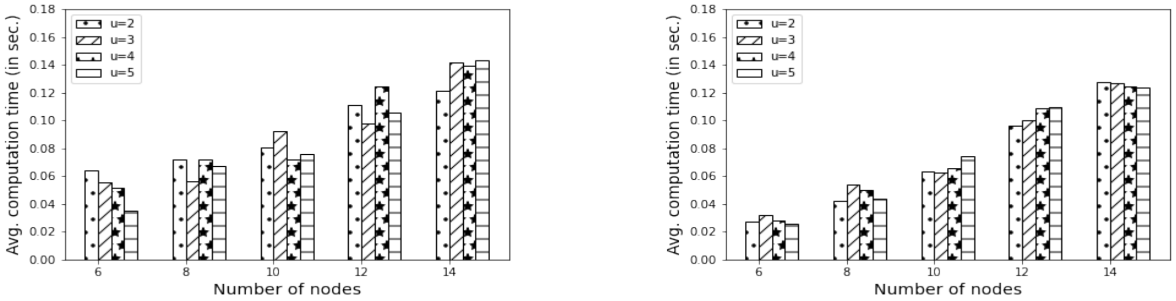
8.2 Comparison of Bayesian Inference Techniques

For comparing the inference algorithms, synthetic graphs are generated as above, but now with a cluster structure [24], where in each



(a) (b)

Fig. 10: Evaluation of computation time using Belief Propagation using Factor Graphs (a) with no evidence, and (b) with evidence.



(a) (b)

Fig. 11: Evaluation of computation time using Variable Elimination (a) with no evidence, and (b) with evidence.

cluster, the number of nodes is the same. Such BAGs with a clustered structure are considered, modeling the ease of vulnerability exploitation within each LAN, broadcast domain, or cluster. Conventionally, there are more intra-cluster access paths in comparison to inter-cluster paths. For each cluster, we generate pseudo-random subgraphs, with a maximum number of parents for each node u. Figs. 15(a) and 15(b) show the average computation time using the inference algorithms, with u=2 and u=5, respectively. Drawing an analogy with the single substation model (Fig. 5), for the synthetic network, 2 clusters with a cluster size of 10 are considered, as shown in Fig. 14. It can be observed that the JT algorithm is faster in comparison to other methods. Table 1 shows the comparison of the inference techniques on the basis of scalability, evidence dependence, computation time, loops, and accuracy.

In our experiments, we alter the number of evidence sources (observable nodes) from 0 to 3, which implies for those nodes, the IDS sends an alert with 100 percent probability. As the number of evidences increases, the computation time for all the methods decreases.

Table 1 Inference method comparison

Technique	Scale	Evi. Dep.	Comp.	Loops	Accuracy
BP_FG	✗	✓	✗	✓	✓
VE	✗	✓	✗	✓	N.A.
PBP-P	✓	✓	✓	✓	✓
JT	✓	✓	✓	✓	N.A.
PBP-T	✓	✓	✓	✗	N.A.

8.3 Evaluation of Inferencing Algorithms on BAGs Generated from Power System Use Cases

The accuracy of the inferred posterior probabilities are calculated based on the root mean square error (RMSE), with reference to an exact inference algorithm such as the JT method. Among the techniques considered in this work, BP_FG and PBP are the approximate inference algorithms. Fig. 16 evaluates the accuracy of the inference techniques for the $8-sub$ dense network case with loops. BP_FG outperforms $PBP-T$ variant of PBP. The BAGs considered for the other use cases without loops have no error in inference. With the increase in evidence, the inference error is reduced for both techniques, except for BP_FG with more error for $evi = 3$ compared to $evi = 2$.

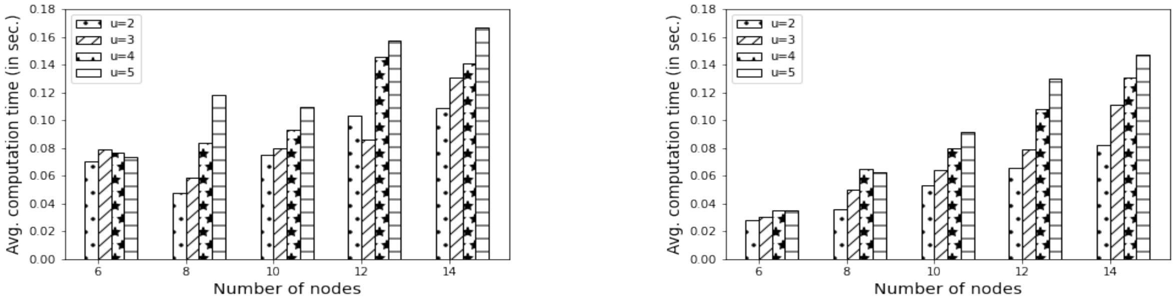
Fig. 17 indicates how the computation time increases logarithmically as the size of the grid increases (from 29 nodes in $1-sub$ case to 217 nodes in $IEEE$ 300 case). $PBP-T$ performs faster than other techniques. It can be observed that the $8-sub$ case cannot be solved using $PBP-T$ due to loops in the BAG. Results show that JT is the preferred technique.

8.4 Evaluation with Varying Threat Intensity

The threat intensity is varied by modifying p_{OR}, as introduced in Section 7. We evaluate how the accuracy of the approximate inference algorithms are affected by altering p_{OR} for the 8-substation case with loops. Table 2 indicates that as the threat intensity increases (increase in p_{OR}), the computation time increases due to increased message passing. The accuracy of inference using BP_FG decreases, while such trend is not observed for PBP. Based on Fig. 16 and Table 2, it can be concluded that BP_FG is preferred over PBP, though the latter technique is not highly dependent on threat intensity. With the highest threat intensity of $p_{OR} = 1$, BP_FG error is still less than PBP.

Table 2 Impact of p_{OR} on the accuracy and computation times of the approximate inference algorithms.

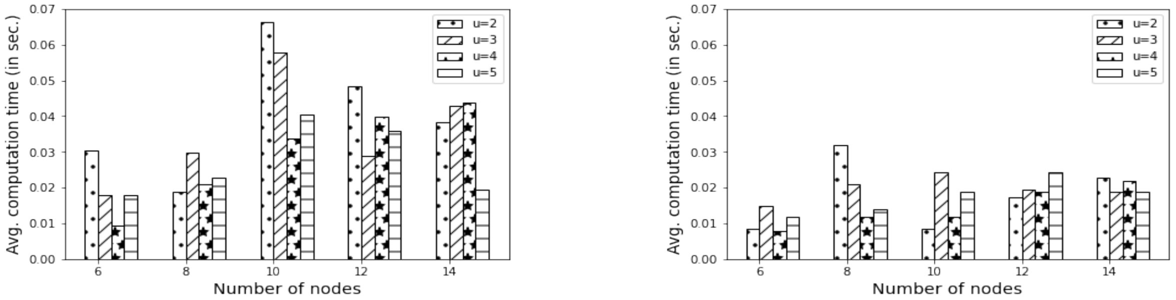
p_{OR}	8-Sub			
	PBP		BP_FG	
	RMSE	Comp. Time	RMSE	Comp. Time
0.0	.0763	.5645	.0027	1.4427
0.2	.0639	.5314	.0056	1.4638
0.4	.0495	.6019	.0079	1.4767
0.6	.059	.5855	.0097	1.5102
0.8	.0467	.6195	.0108	1.7923
1.0	.0429	.7113	.0121	1.8869



(a)

(b)

Fig. 12: Evaluation of computation time using Pearl's Belief Propagation Algorithm (a) with no evidence, and (b) with evidence.



(a)

(b)

Fig. 13: Evaluation of computation time using Junction Tree with (a) no evidence, and (b) with evidence.

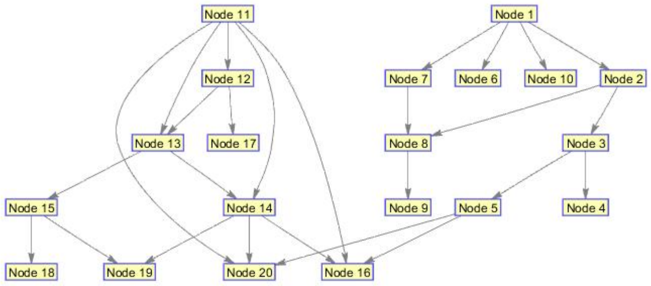


Fig. 14: A clustered Bayesian Attack Graph with 2 clusters and 10 nodes in each cluster.

8.5 Feature-based BAG Structure Learning

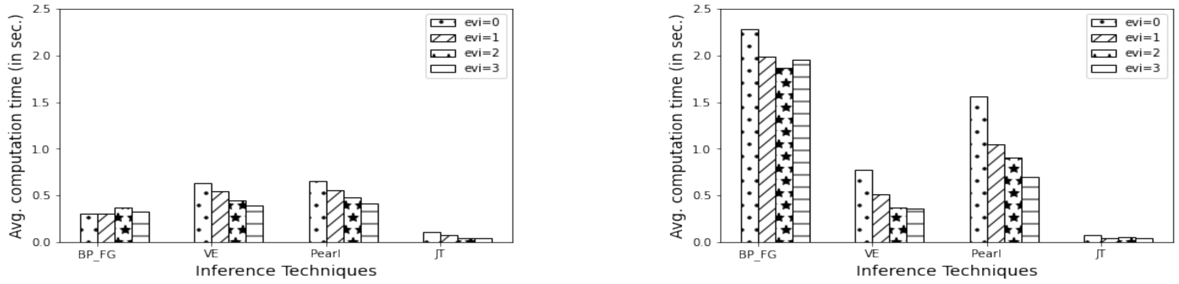
This section analyzes the structure learning problem, leveraging RESLab that emulates the MiTM attacks, as introduced in Section 5.1 and 5.2.5. The cyber and physical features extracted from the emulation, presented in [46], form the nodes in the BAG. The relationships, or causal links, between the nodes are learned using the Chow Liu and K2 algorithms in [10]. The experiment is performed on the use case where the measurements are tampered [38]. First, two physical measurements, the real power generation values in MW of two generators that were targeted in the $UC3$ MiTM attack, are extracted from three different locations in the emulated network: the node where DNP3 master runs (M), the substation router (R), and the node that runs PWDS, i.e., DNP3 outstations (O) (Fig. 18), making a six-node BAG (two measurements per location). During the attack, the measurement the router receives from the intruder spoofing the outstation is a modified value, compared to that sent by the benign outstation. Hence, the learned structure for the dataset with attack must reduce the correlation between node O and R for both the measurements, i.e., it either removes the link or reduces the conditional probability $P(R|O)$ in comparison to the dataset with no compromise. The structure learned from the Chow-Liu algorithm (Fig. 18) is $(R1, M1), (M1, O1), (M1, O2), (O2, R2), (R2, M2)$, while from the K2 algorithm is $(O1, M1), (R1, M1), (R1, O1), (R2, M2)$. The 1 and 2 indices in the BAG nodes denote the first and second generator identifiers, respectively. Since both $(R1, M1)$ and $(R2, M2)$ exist in both solutions, it clearly shows that the intruder is in the substation LAN, i.e., modification of a measurement in the DNP3 packet occurred before the packet left the substation router R.

Next, cyber features of source and destination MAC address from the three location are considered, making a six-node BAG. The structure learned from the Chow-Liu algorithm is $(O1, R1)$, $(O1, R2)$, $(R2, O2)$, $(R2, M1)$, $(M1, M2)$ and from the K2 algorithm is $(M1, M2)$, $(O2, R2)$, $(O2, O1)$, $(R2, O1)$, $(O1, R1)$. The 1 and 2 indices denote the source and destination MAC address, respectively. In the ARP spoof attack, the packet received at the router from the intruder will have a different source MAC address from the MAC address of the outstation. Similarly, the packet received at the outstation from the intruder will have a different source MAC address from the MAC address of the router. Existence of $O1, R1$ in both the techniques indicate that both the router and the outstation received packets from the intruder.

The existence of some unwanted links can be reduced by collecting a dataset from emulation over a longer period. From the law of large numbers, the average of the results from a large number of trials should be close to the expected value. Hence, computing the mean of the prior distribution, with less variance, will improve with more data.

9 Conclusion

Better modeling and visibility of dynamic adversary behavior can improve power system attack-resilience. The major contributions of this paper are generation of BAGs, analysis and comparison of inference algorithms based on scale, evidence dependency, time complexity, accuracy and loops, for different power system use-cases. Some major conclusions drawn are the following: Belief propagation with factor graphs is computationally expensive for BAGs with high average in-degrees; hence, it can only work when nodes have fewer vulnerabilities and in smaller networks. Junction tree based inference outperformed the other three techniques based on its low



(a)

(b)

Fig. 15: (a) Comparison of inference techniques with maximum allowed parents (u = 2), with varying no. of observed nodes; (b) Comparison of inference techniques with maximum allowed parents (u = 5), with varying no. of observed nodes.

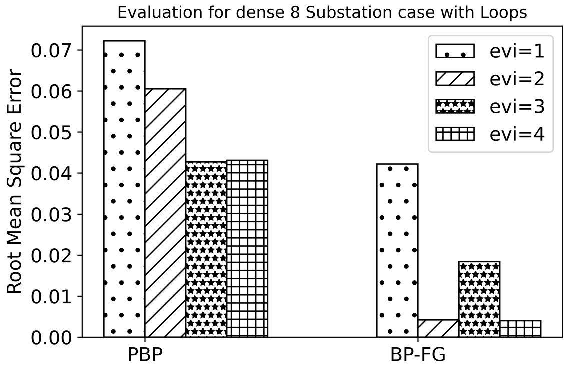


Fig. 16: Error evaluation of the approximate inference algorithms.

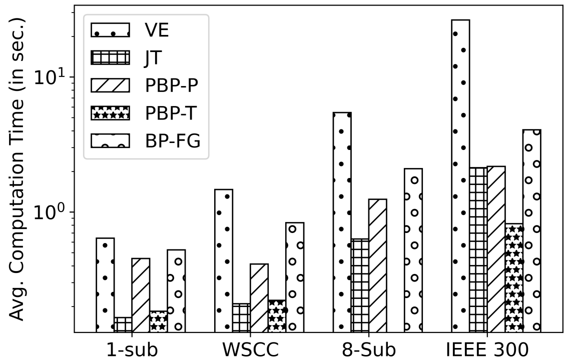


Fig. 17: Inference computation times for four power system cases.

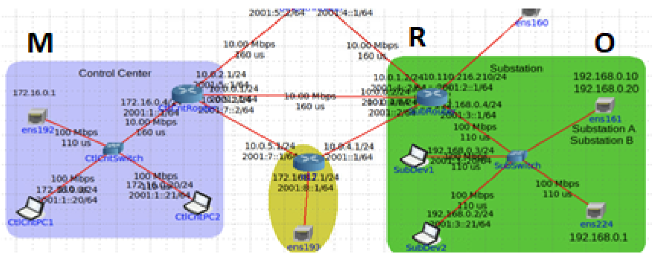


Fig. 18: Testbed configuration, with measurements and features extracted from the DNP3 master (M), a substation router (R), and a DNP3 outstation (O).

computation time and scalability to large networks. Evidence plays a major role in effecting the complexity of the inference algorithms. Hence, the inference algorithms should be considered based on the sensitivity of the IDSs to intrusions. VE and JT methods should be used for exact inference, and PBP can be used for approximate inference in loopy BAGs. Accuracy of the approximate inference algorithms depends on the network density and the attack strength, regulated through p_{OR}. BP_FG should be considered for higher accuracy and $PBP-P$ for faster inference. Finally, a Bayesian framework is developed within a cyber-physical EMS for evaluation and comparison of inference algorithms for different power system use cases.

Future work remains on collecting the information in reality, as the exact estimate of the complexity and cost associated with this approach for a given utility remains an open challenge to be better understood. What is known is that utilities are moving in this direction, and there are a lot of existing data sources that are already being collected, with these objectives in mind. Hence, this work intends to help electric power utilities do this work in their own systems, as it can seed new defense toolsets that domain experts can advise stakeholders toward real implementation in practice. Bayes-CAPS is currently part of the suite of CYPRES prototype tools. The current work provides the details of the approach and implementation, including the pseudocode and algorithms. Future work will look to create a standalone opensource tool and data archive.

10 Acknowledgements

This research is supported by the US Department of Energy's (DOE) Cybersecurity for Energy Delivery Systems program under award DE-OE0000895.

11 References

1 N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, vol. 5, p. 6, 2011.

2 E-ISAC, "Analysis of the cyber attack on the ukrainian power grid defense use case."

3 E. Targett. (2020, March) High Voltage Attack: EU's Power Grid Organisation Hit by Hackers. [Online]. Available: https://www.cbronline.com/news/eu-power-grid-organisation-hacked

4 J. Wang, K. Fan, W. Mo, and D. Xu, "A method for information security risk assessment based on the dynamic bayesian network," in *2016 International Conference on Networking and Network Applications (NaNA)*, 2016, pp. 279–283.

5 S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15*, 2002, pp. 49–63.

6 Y. Li, J. Chen, and L. Feng, "Dealing with uncertainty: A survey of theories and practices," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2463–2482, 2013.

7 A. Sahu and K. Davis, "Inter-domain fusion for enhanced intrusion detection in power systems: An evidence theoretic and meta-heuristic approach," *Sensors*, vol. 22, no. 6, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/6/2100

8 FireEye, "Shining a light on darkside ransomware operations," May 2021. [Online]. Available: https://www.fireeye.com/blog/threat-research/2021/05/shining-a-light-on-darkside-ransomware-operations.html

9 N. Vlassis, M. Ghavamzadeh, S. Mannor, and P. Poupart, *Bayesian Reinforcement Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 359–386. [Online]. Available: https://doi.org/10.1007/978-3-642-27645-3_11

10 A. Sahu and K. Davis, "Structural learning techniques for bayesian attack graphs in cyber physical power systems," in *2021 IEEE Texas Power and Energy Conference (TPEC)*, 2021, pp. 1–6.

11 Peng Xie, J. H. Li, Xinming Ou, Peng Liu, and R. Levy, "Using bayesian networks for cyber security analysis," in *2010 IEEE/IFIP International Conference on Dependable Systems Networks*, 2010, pp. 211–220.

12 W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613000042

13 N. Gaudet, A. Sahu, A. E. Goulart, E. Rogers, and K. Davis, “Firewall configuration and path analysis for smartgrid networks,” in *2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, 2020, pp. 1–6.

14 K. Davis, “An energy management system approach for power system cyber-physical resilience,” in *invited position paper for 2021 Virtual Workshop on Cyber Experimentation and Science of Security (CESoS))*, Nov 2021.

15 A. Sahu, H. Huang, K. Davis, and S. Zonouz, “A framework for cyber-physical model creation and evaluation,” in *2019 20th International Conference on Intelligent System Application to Power Systems (ISAP)*.

16 P. Wlazlo, K. Price, C. Veloz, A. Sahu, H. Huang, A. Goulart, K. Davis, and S. Zounouz, “A cyber topology model for the texas 2000 synthetic electric power grid,” in *2019 Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2019, pp. 1–8.

17 T. Le and C. N. Hadjicostis, “Graphical inference for multiple intrusion detection,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 370–380, 2008.

18 M. Frigault and L. Wang, “Measuring network security using bayesian network-based attack graphs,” in *2008 32nd Annual IEEE International Computer Software and Applications Conference*, 2008, pp. 698–703.

19 F. Jemili, M. Zaghdoud, and M. B. Ahmed, “A framework for an adaptive intrusion detection system using bayesian network,” in *2007 IEEE Intelligence and Security Informatics*, 2007, pp. 66–70.

20 P. Wu, Y. Shuping, C. Junhua, and W. Zhigang, “Recognizing intrusive intention based on dynamic bayesian networks,” in *2009 International Symposium on Information Engineering and Electronic Commerce*, 2009, pp. 241–244.

21 O. J. Mengshoel, M. Chavira, K. Cascio, S. Poll, A. Darwiche, and S. Uckun, “Probabilistic model-based diagnosis: An electrical power system case study,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 5, pp. 874–885, 2010.

22 B. Delfino, G. B. Denegri, M. Invernizzi, A. Canonero, and P. Forzano, “An expert system for alleviating overloads in electric power systems: general concepts and applications,” in *[1988] Proceedings. The Fourth Conference on Artificial Intelligence Applications*, 1988, pp. 299–304.

23 A. Peerzada, M. Begovic, W. Rohouma, and R. Balog, “On estimation of equipment failures in electric distribution systems using bayesian inference,” in *2021 54th Hawaii International Conference on System Sciences*, 2021, pp. 3131–3140.

24 L. Muñoz-González, D. Sgandurra, M. Barrère, and E. C. Lupu, “Exact inference techniques for the analysis of bayesian attack graphs,” *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 231–244, 2019.

25 W. Ren, T. Yu, T. Yardley, and K. Nahrstedt, “Captar: Causal-polytree-based anomaly reasoning for scada networks,” in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–7.

26 M. Aksu, K. Bicakci, M. H. Dilek, M. Ozbayoglu, and E. Tatlı, “Automated generation of attack graphs using nvd,” 03 2018, pp. 135–142.

27 National Institute of Standards and Technology (NIST), “National vulnerbality database.” [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss

28 Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, “A practical approach to constructing a knowledge graph for cybersecurity,” *Engineering*, vol. 4, no. 1, pp. 53 – 60, 2018, cybersecurity. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2095809918301097

29 Xueqiu, Q. Jia, S. Wang, C. Xia, and L. Lv, “Automatic generation algorithm of penetration graph in penetration testing,” in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2014, pp. 531–537.

30 S. Zhang, L. Li, J. Li, S. Song, and X. Chen, “Using attack graphs and intrusion evidences to extrapolate network security state,” in *2009 Fourth International Conference on Communications and Networking in China*, 2009, pp. 1–6.

31 Sungmin Jung, Gyubok Moon, Yongjun Kim, and Kyungwhan Oh, “Planning based on dynamic bayesian network algorithm using dynamic programming and variable elimination,” in *2009 4th International Conference on Autonomous Robots and Agents*, 2009, pp. 109–114.

32 P. G. Bringas, “Intensive use of bayesian belief networks for the unified, flexible and adaptable analysis of misuses and anomalies in network intrusion detection and prevention systems,” in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*, 2007, pp. 365–371.

33 Y. Huangfu, L. Zhou, and C. Yang, “Routing the cyber-attack path with the bayesian network deducing approach,” in *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2017, pp. 5–10.

34 F. R. Kschischang, B. J. Frey, and H. . Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

35 R. Dechter, “Bucket elimination: A unifying framework for reasoning,” *Artificial Intelligence*, vol. 113, no. 1, pp. 41 – 85, 1999.

36 J. Yedidia, W. Freeman, and Y. Weiss, *Understanding belief propagation and its generalizations*, 01 2003, vol. 8, pp. 239–269.

37 R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*, 01 2001, vol. 43.

38 A. Sahu, P. Wlazlo, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, “Design and evaluation of a cyber-physical resilient power system testbed,” 11 2020. [Online]. Available: http://arxiv.org/abs/2011.13552

39 A. Ankan, “pgmpy-python library for probabilistic graphical models.” [Online]. Available: https://github.com/pgmpy/pgmpy

40 Microsoft, “Infer.net,” 2008. [Online]. Available: https://dotnet.github.io/infer/

41 H. Nguyen, K. Palani, and D. Nicol, “An approach to incorporating uncertainty in network security analysis,” 04 2017, pp. 74–84.

42 Y. Liu and H. Man, “Network vulnerability assessment using bayesian networks,” *Proc SPIE*, 03 2005.

43 M. Frigault, L. Wang, A. Singhal, and S. Jajodia, “Measuring network security using dynamic bayesian network.” Association for Computing Machinery, 2008.

44 P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, “Man-in-the-middle attacks and defence in a power system cyber-physical testbed,” *IET Cyber-Physical Systems: Theory & Applications*, vol. 6, no. 3, pp. 164–177, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cps2.12014

45 M. R. Narimani, H. Huang, A. Umunnakwe, Z. Mao, A. Sahu, S. Zonouz, and K. Davis, “Generalized contingency analysis based on graph theory and line outage distribution factor,” *IEEE Systems Journal*, vol. 16, no. 1, pp. 626–636, 2022.

46 A. Sahu, Z. Mao, P. Wlazlo, H. Huang, K. Davis, A. Goulart, and S. Zonouz, “Multi-source multi-domain data fusion for cyberattack detection in power systems,” *IEEE Access*, vol. 9, pp. 119 118–119 138, 2021.

47 ——, “Cyber-physical dataset for mitm attacks in power systems,” *IEEE Dataport*, 2021.

48 G. A. Weaver, K. Davis, C. M. Davis, E. J. Rogers, R. B. Bobba, S. Zonouz, R. Berthier, P. W. Sauer, and D. M. Nicol, “Cyber-physical models for power grid security analysis: 8-substation case,” in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2016, pp. 140–146.

49 Cyber Physical Resilient Energy Systems, “Test cases.” [Online]. Available: https://cypres.engr.tamu.edu/test-cases/

50 K. R. Davis, R. Berthier, S. Zonouz, G. Weaver, R. B. Bobba, E. Rogers, and D. M. N. P. W. Sauer, “Cyber-physical security assessment for electric power systems,” in *IEEE-HKN: The Bridge*, 2016.

51 K. R. Davis, C. M. Davis, S. A. Zonouz, R. B. Bobba, R. Berthier, L. Garcia, and P. W. Sauer, “A cyber-physical modeling and assessment framework for power grid infrastructures,” *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2464–2475, 2015.

52 A. Sahu, “CyPSA-Live Code,” Nov. 2022. [Online]. Available: https://github.com/Abhijeet1990/CYPSA-Live

53 Z. Mao, A. Sahu, P. Wlazlo, Y. Liu, A. Goulart, K. Davis, and T. J. Overbye, “Mitigating tcp congestion: A coordinated cyber and physical approach,” in *2021 North American Power Symposium (NAPS)*, 2021, pp. 1–6.

54 Illinois Center for a Smarter Electric Grid, “WSCC 9-bus system.” [Online]. Available: https://icseg.iti.illinois.edu/wscc-9-bus-system/

55 A. Umunnakwe, A. Sahu, M. R. Narimani, K. Davis, and S. Zonouz, “Cyber-physical component ranking for risk sensitivity analysis using betweenness centrality,” *IET Cyber-Physical Systems: Theory & Applications*, vol. 6, no. 3, pp. 139–150, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cps2.12010

56 A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, “Grid structural characteristics as validation criteria for synthetic networks,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258–3265, 2017.

57 N. Poolsappasit, R. Dewri, and I. Ray, “Dynamic security risk management using bayesian attack graphs,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.

58 F. Girlich, M. Rossberg, and G. Schaefer, “On the resistance of overlay networks against bandwidth exhaustion attacks,” *Telecommunication Systems*, vol. 60, 03 2015.

59 K. Murphy, “How to use the bayes net toolbox,” 2007. [Online]. Available: http://bayesnet.github.io/bnt/docs/usage.html