Journal Pre-proof

Cybersecurity vulnerability mitigation framework through empirical paradigm: Enhanced prioritized gap analysis

Sri Nikhil Gupta Gourisetti, Michael Mylrea, Hirak Patangia

PII: S0167-739X(19)30734-4
DOI: https://doi.org/10.1016/j.future.2019.12.018
Reference: FUTURE 5333

To appear in: *Future Generation Computer Systems*

Received date : 16 March 2019
Revised date : 1 November 2019
Accepted date : 11 December 2019

Please cite this article as: S.N.G. Gourisetti, M. Mylrea and H. Patangia, Cybersecurity vulnerability mitigation framework through empirical paradigm: Enhanced prioritized gap analysis, *Future Generation Computer Systems* (2019), doi: https://doi.org/10.1016/j.future.2019.12.018.

Cybersecurity Vulnerability Mitigation Framework through Empirical Paradigm: Enhanced Prioritized Gap Analysis

Sri Nikhil Gupta Gourisetti[1,2], Michael Mylrea[1,2], Hirak Patangia[2]

[1]Pacific Northwest National Laboratory, Richland, WA 99354 USA
[2]Engineering Sciences and Systems (Electrical and Computer Engineering), University of Arkansas, Little Rock, AR 99354 USA
Corresponding author: Sri Nikhil Gupta Gourisetti (e-mail: srinikhil.gourisetti@pnnl.gov; gigupta@ualr.edu).
Declarations of interest: none

Abstract Existing cybersecurity vulnerability assessment tools were designed based on the policies and standards defined by organizations such as the U.S. Department of Energy and the National Institute of Standards and Technology (NIST). Frameworks such as the cybersecurity capability maturity model (C2M2) and the NIST Cybersecurity Framework (CSF) are often used by the critical infrastructure owners and operators to determine the cybersecurity maturity of their facility. Although these frameworks are exceptional at performing qualitative cybersecurity analysis and identifying vulnerabilities, they do not provide a means to perform prioritized mitigation of those vulnerabilities in order to achieve a desired cybersecurity maturity. To address that challenge, we developed a framework and software application called the cybersecurity vulnerability mitigation framework through empirical paradigm (CyFEr). This paper presents the detailed architecture of CyFEr's enhanced prioritized gap analysis (EPGA) methodology and its application to CSF. The efficacy of the presented framework is demonstrated by comparing against existing similar models and testing against the cyber injects from a real-world cyber-attack that targeted industrial control systems (ICS) in critical infrastructures.

Index Terms cybersecurity vulnerability assessment; cybersecurity framework; cybersecurity mitigation; criteria ranking.

1. INTRODUCTION

The digital transformation of the world has been on the rise due to the growing number of smart devices, precise high-speed sensors, and various classes of networked systems that are often referred to under the umbrella of Internet-of-Things (IoT) [1]. As part of the ongoing technological advancements, critical infrastructure facilities have been adopting networked smart devices (often referred to as Industrial IoT, or IIoT) to streamline various activities, including autonomous control of industrial control systems (ICS), decentralized and advanced sensing and communication technologies [2], and integration of machine learning and artificial intelligence for precise data analytics. According to a statistical projection [3], the penetration of such smart devices across the global infrastructure is expected to grow from a current value of 26 billion devices to almost 75 billion devices by 2025. Following a similar trend, critical infrastructure automation systems are expected to grow between 11%–20% by 2022–2026 [4] [5]. Such a rise in the use of networked smart devices is leading to an expanding cybersecurity threat landscape, and the exacerbation of the emerging cyber threats and security challenges cannot be overstated [6].

In the present expanding cybersecurity threat landscape, in order for the critical infrastructure organizations to protect their network of smart devices, it is essential for them to assess their infrastructure's security posture and discover vulnerabilities. In order for them to do that effectively, the critical infrastructure owners should be equipped with reliable tools and applications, such as cybersecurity frameworks and maturity models, to assess their overall cybersecurity posture. Cybersecurity frameworks and maturity models have the ability to facilitate the organizational owners and operators to identify and protect their critical systems from evolving cyber threats while also detecting cyber intrusions that could cause disruption to operations. In addition, the organizations should be equipped with capabilities to respond and recover under a critical cyber event. All of these combined factors and processes will potentially enable an organization to perform risk-informed decisions with critical safeguards in place. This paper demonstrates a methodology and tool that will facilitate the critical infrastructure owners and operators to develop autonomous vulnerability mitigation strategies and associated policies in order to withstand and counter sustained cyber-attacks.

In order to address the many cybersecurity policy compliance challenges in the converged information technology (IT) and operational technology (OT) networks, researchers have attempted to create a variety of vulnerability assessment tools [7] [8], methodologies [9], and frameworks [10] [11] [12]. However, most of these methodologies address the challenges at a systems level for specific applications, and therefore, lack scalability and adaptability. Among the plethora of cybersecurity tools and frameworks, the cybersecurity capability maturity model (C2M2) [13] designed by the U.S. Department of Energy (DOE) and the cybersecurity framework (CSF) designed by the National institute of Standards and Technology (NIST) [14] have gained a positive reputation. In addition, they do not appear to be as constrained as many of the other maturity model-based vulnerability assessment tools.

The core architectures of both CSF and C2M2 are designed so they can be adopted to any particular application that focuses primarily on compliance with cybersecurity policies, standards, and regulations. Although both C2M2 and CSF provide a detailed analysis of the identified cybersecurity vulnerabilities in a critical organization, they do not appear to have a mechanism that can be used by the

critical infrastructure owners to prioritize and develop vulnerability mitigation plans and procedures. To address that challenge, a novel quantitative prioritized vulnerability assessment mechanism and a web-based software application are developed to ingest the core critical elements of the CSF and provide an autonomous mitigation of cybersecurity vulnerabilities. This is achieved by coupling multi-criteria decision analysis (MCDA) techniques [15] with weighted dependency structures.

One of the major challenges in cybersecurity research is the availability of data. Since the organizations seldom share findings, such as system and network vulnerabilities, developing machine learning applications and other adaptive software is challenging. In similar scenarios where data is constrained [16] [17] [18], researchers often use rank-weight methods combined with layered logical constructs. However, these techniques have not been extensively used for cybersecurity vulnerability assessments.

In the presented research, the cybersecurity controls from CSF are combined with MCDA techniques and multi-tiered mathematical filters to perform a constrained security analysis. This process resulted in a hierarchical outcome that the critical infrastructure owners and operators can follow in order to mitigate cybersecurity vulnerabilities. The presented application is data agnostic, scalable, and can also be adopted to frameworks and methodologies beyond CSF.

The novelty of the proposed framework lies in its ability to combine quantitative ranking techniques with dependency structures to perform enhanced prioritized vulnerability mitigation. The novel mitigation process presented in this paper is based on logical constructs and multi-tiered mathematical filters (discussed in later sections). The proposed framework can be used at the application-level, system-level, or organizational and management level. Therefore, it is compatible with several existing maturity and vulnerability analysis models, such as C2M2, CSF, RMF, CSET, NICE-CMM [19], CRI [20], CPI [21], ISM [22] [23], and many more. The core innovation of the proposed framework lies in its three-part process: Part-1: design the dependency structures; Part-2: calculate the priority stages and masses; and Part-3: perform an EPGA. To construct the dependency trees, a mathematical relationship is defined across a set of complex cybersecurity controls that are grouped under a set of pre-defined criteria. The criteria are defined strictly based on the domains and subdomains depicted in the NIST CSF. After designing the dependency trees, the relative masses of the controls are calculated based on one of three methods: 1) relative mass-based calculation; 2) stage, maturity indicator level (MIL), relative mass-based calculation (MIL is a numerical designation associated with a cybersecurity control that defines the criticality, importance, complexity, and associated inter-dependencies of a cybersecurity control); or 3) MIL and relative mass-based calculation. Upon calculating the masses, EPGA is performed. EPGA is an eight-step process that incorporates goal identification; rank-based criteria weight calculation; multi-constrained based solution filtration; enhanced present state factor, implementation state factor, and transition state factor calculation; performance score index; and weighted performance score calculation. This process will produce prioritized vulnerabilities to be mitigated in order to reach a targeted cybersecurity maturity.

The efficacy of the presented framework and methodology is demonstrated by testing against a major real-world cyber-attack that targeted critical infrastructure ICS and resulted in a significant loss of property. The results obtained through the presented framework are compared against traditional subjective methods to show the accuracy, adaptability, and ease-of-use of the presented framework. This paper is organized as follows: Section – 2 provides a technical background of the fundamental research used in the presented framework, Section – 3 presents the core mathematical model of EPGA including all the underlying formulation and logical constructs, Section – 4 demonstrates the efficacy of EPGA through a use-case, Section – 5 performs an extensive applicability and comparative analysis where EPGA is compared against several of the existing frameworks & models, and Section – 6 concludes the paper.

2. Literature Review

The objective of this section is to provide a detailed overview of the existing cybersecurity methodologies and frameworks that fall into the following categories: 1) cybersecurity applications that adopted MCDA methods and 2) cybersecurity methodologies and frameworks that performed prioritized vulnerability mitigation. Through this section, we will clearly demonstrate the difference between the existing methods and the proposed EPGA method.

2.1. Existing Applications and Limitations of the MCDA Methods in Cybersecurity Research

In this sub-section, we present an overview of the well-known cybersecurity models that use MCDA, their limitations, and the potential relationship with our proposed solution. Cybersecurity vulnerability assessments through maturity models involve analyzing a network, system, or facility through a set of security controls. The number of security controls can range from 100–500+, depending on which tool or framework is used (e.g., C2M2, CSF, RMF). Therefore, it is important to have a comprehensive list of prioritized criteria in order to determine the ideal cybersecurity posture; based on this complexity, the problem at hand is an MCDA problem.

When using MCDA, specific weights are assigned to a set of criteria and then normalized to a scale of 1 or 100 before proceeding to the decision-making process. Such a subjective weighting process often adds more complexity to an already complex problem [24] due to the ambiguity in how to assign weights to the criteria and how to choose weights for the individual criteria. This can be simplified by ranking the criteria. Standard rank-weight methods can be used to design a computational algorithm to use the ranks and calculate the relative weights. Many rank-weight

methods have been rigorously analyzed and used in various research areas [25] [26], but each of these methods may perform differently depending on the intended use.

Although the rank-weight methods have been used in other areas, such as logistics, inventory management, and economics/cost-benefit analysis, the combination of rank-weight methods and MCDA techniques in cybersecurity vulnerability mitigation has not been explored by many researchers. In one of the recent attempts to use rank-weight methods in cybersecurity [27], the authors demonstrated a quantitative risk assessment methodology. Through their work, the authors were able to relatively estimate the risk value of various use cases based on the subjective estimates and assumptions about the cyber threats, vulnerabilities, and consequences.

Another similar risk estimation attempt based on partial-MCDA techniques was presented in [28]. In [28], the authors evaluated various smart grid systems to grade a small set of high-level vulnerabilities, likelihood of attack, and impact to qualitatively estimate relative risk value. Other risk quantification research publications and government reports that focused on likelihood, impact, and associated risk value are detailed in [29] [30] [31] [32] [33]. Although, those methods are efficient in performing relative risk estimation, they have certain limitations and gaps that restrict their scalability and the ability to use them towards vulnerability mitigation, including the following: 1) All of the previously mentioned methods assumed a high-level set of organizational-level vulnerabilities and developed their formulation based on *manual* grading/scoring methods. Therefore, adopting them to application-level and system-level maturity models, such as the CSF and C2M2, with 100+ security controls is a herculean task. 2) They do not facilitate the prioritization of the vulnerabilities to mitigate. Therefore, obtaining the overall organizational risk estimates may not fully facilitate the development of vulnerability mitigation plans. However, coupling those methods with a vulnerability mitigation method, such as the presented method, can define a path to reduce the risk associated with various systems in the critical infrastructure facilities. 3) Due to the level of subjectivity associated with the methods, the end result has very minimal to no mathematical or data-driven support. Therefore, validating the outcomes of the approaches based on logical constructs may not be possible. Realizing the gaps and limitations in some of the well-vetted methods, we attempted to use their base process as a fundamental starting point to develop the proposed MCDA-based vulnerability mitigation method.

2.2. Existing Methods: Prioritization of Cybersecurity Attributes

In addition to the above discussed methods and frameworks, some researchers have been exploring ways to prioritize cybersecurity attributes, such as risks, consequences, and vulnerabilities. In this sub-section, we will provide an overview of the methods, identify the gaps in those methods, analyze any potential relationship between those methods and EPGA, and clearly show the unique value proposition of EPGA.

The research performed by Gadyatskaya et al. [34] demonstrates a process to rank the cyber-attack scenarios and nodes using a software called ADTool 2.0. In [34], the researchers adapted the attack trees to evaluate the cyber-attack scenarios at every node of the attack tree. Then, each node is ranked with respect to those cyber-attack scenarios. It is indicative that the researchers made an inherent assumption that the nodes were vulnerable to those attack scenarios. Therefore, since the method described in [34] is not related to vulnerability assessment, that method may not be applicable for identification and prioritization of vulnerabilities. During our review of [34], we made an interesting observation—since EPGA is designed to ingest a set of prioritized attributes (such as the identified vulnerabilities), ADTool 2.0 can potentially feed its output to EPGA to further prioritize the nodes based on the prioritized attack scenarios and nodes. According to Gadyatskaya et al., a large complicated set of attack scenarios and nodes could make it a non-trivial process to rank the scenarios and nodes, let alone prioritize them for targeted resource allocation. Therefore, EPGA can potentially address some of those challenges related to prioritization.

Lundberg et al. published a research article [35] to better explain the prioritized risks, consequences, and associated security strategies. The document articulated the high-level challenges associated with weighted prioritization of hazards and threats. Those include 1) individual biases where certain aspects of the risks can potentially distort one's perception of risk [36] [37] [38] [39] [40], 2) disagreements and differing opinions in terms of estimating the value of risks and consequences, and 3) the non-trivial subjective process of assigning weights and performing numerical estimates [41] [42] [43] [44]). A similar observation was noted by [35], highlighting the difficulty and impracticality associated with the weight-based prioritization. However, the weight-based quantification can be effective if the disagreement among the evaluators who assign weights is resolved. The authors of [35] also indicated the effectiveness of the use of rank-weight methods to mitigate the challenge of weight-based disagreement. Those methods were discussed at great detail in [45] [46] [47] [48] [49] [50]. It is clear that the above challenges can be relatable to the vulnerability prioritization problem that we are addressing in this paper. Therefore, as shown in the later sections of this paper, the EPGA technique prioritized the attributes through a relative ranking method (instead of weighted approach). In addition, [35] discussed various parameters and recommendations [50] [51] to perform attribute ranking. However, [35] did not demonstrate the adoption of those processes to solve the vulnerability prioritization problem. Therefore, the processes defined in [35] can be used in conjunction with the rank-weight methods described in the next section to ensure effective attribute ranking and vulnerability mitigation. It is

evident that one of the objectives of [35] was to evaluate various ranking processes with respect to hazards and associated risks. In order to address the risks, it is important to understand the vulnerabilities. Therefore, by custom fitting the processes defined in [35] towards vulnerability prioritization and mitigation, those processes can be clearly used as precursors and as post-processes to the EPGA method demonstrated in this paper.

Beyond risk and consequence prioritization, researchers also developed methods to prioritize asset-level vulnerabilities [52] [53] [54] [55] [56]. In this context, the term asset may refer to a hardware system or a software system or firmware. In [52], researchers used the common vulnerability scoring system (CVSS) [57] and common vulnerabilities and exposures (CVE) [58] to develop a severity ranking method and to prioritize the discovered asset-level vulnerabilities. A similar CVSS-related method is shown in [53]. In [53], the research performed an asset-level analysis (similar to [52]) with a possible assumption that the organizational vulnerabilities are mitigated (or that assumption/factor may not have been considered). Such an assumption may be acceptable in the case of [53] because the objective was to address the vulnerabilities related to specific assets. In [54], the researchers used a CVSS of the asset-level vulnerabilities as the basis to rank those vulnerabilities and to mitigate them. In [55], the researchers combined the CVSS with an MCDA technique called the multiplicative analytic hierarchy process (MAHP) [59] [60] to perform a prioritized mitigation of the asset vulnerabilities. In their approach, the discovered vulnerabilities were used as a critical segment to define the criteria in the MCDA problem. Finally, [56] demonstrated a similar CVSS-based method focusing on the asset-level vulnerability prioritization and mitigation. Based on the analysis of [52] [53] [54] [55] [56], it is evident that all of the methods used the existing CVSS of known vulnerabilities as the fundamental basis. It is important to note that the CVE and CVSS are always associated with asset-level vulnerabilities, and they are not defined with respect to the organizational cybersecurity. The vulnerability assessment tools, such as CSF and C2M2, are developed to address the organizational cybersecurity problems related to the policies and procedures, network strategies, and compliance with cybersecurity standards. The research demonstrated in [52] [53] [54] [55] [56] can be potentially used to address asset-level vulnerability mitigation, but they cannot be used to address organizational cybersecurity threats and vulnerabilities. As observed in [52] [53] [54] [55] [56], the researchers used CVSS and then manually quantified the prioritization of the vulnerabilities. Relative to an entire organization, an asset may have a small number of vulnerabilities. Therefore, the processes defined in [52] [53] [54] [55] [56] may be acceptable and practical when addressing the vulnerabilities of an asset or a set of assets. However, those processes cannot be adapted to address organizational cybersecurity vulnerabilities. Therefore, the clear difference between those methods and EPGA is that the EPGA method is developed to integrate with tools such as CSF, C2M2, RMF, and others to develop a prioritized mitigation plan for the organizational cybersecurity vulnerabilities. In addition, there is a clear potential to use EPGA as a post-processing step to the methods defined in [52] [53] [54] [55] [56]. Such integration may be very effective if a large number of assets are evaluated.

3. Enhanced Prioritized Gap Analysis

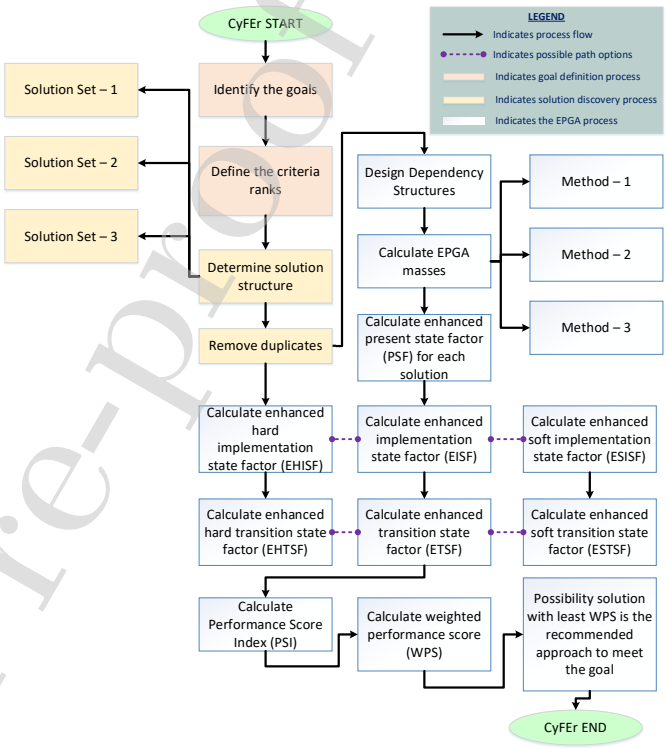


FIGURE 1. CyFEr workflow diagram with an emphasis on EPGA

The nature of CSF is to categorize controls into three MILs. The C2M2 and CSF defines four MILs, MIL0 through MIL3, that apply independently to each of their domains. 1) The MILs apply independently to each domain. As a result, an organization may be operating at different MIL ratings for different domains. For example, an organization could be operating at MIL1 in one domain, MIL2 in another domain, and MIL3 in a third domain. 2) The MILs are cumulative within each domain; to earn an MIL in a domain, an organization must perform all of the practices in that level and its predecessor level(s). For example, an organization must perform all of the domain practices in MIL1 and MIL2 to achieve MIL2 in the domain. Similarly, the organization would have to perform all practices in MIL1, MIL2, and MIL3 to achieve MIL3. 3) Establishing a target MIL for each domain is an effective strategy for using the C2M2 and CSF to guide cybersecurity program improvement. Organizations should become familiar with the practices in the C2M2 and CSF prior to determining target MILs. Gap analysis and improvement efforts should then focus on achieving those target levels. 4) Practice performance and MIL achievement need to align with business objectives and the organization's cybersecurity strategy. Striving to achieve the highest MIL

in all domains may not be optimal. Organizations should evaluate the costs of achieving a specific MIL against potential benefits.

However, they do not define interconnected dependencies between those cybersecurity controls. The CSFs identify vulnerabilities, but do not provide an analysis to prioritize the vulnerabilities. The following sections are focused on designing dependency trees for the control structure and connecting them to the EPGA core formulation; the overall CyFEr workflow diagram, with an emphasis on EPGA, is shown in FIGURE 1. Below is the sequence of steps as defined in FIGURE 1. Details of the following steps are discussed in the following sections.

1. *Nomenclature:* All solid arrows are the mandatory steps of CyFEr. Some of these steps may involve critical infrastructure owners (users) intervention. Each dotted arrow indicates that the user should choose one of the options. Each set of major phases are shown by a unique color. Initial user input phase is shown by orange boxes; optimization and solution filtration process is shown as yellow; the core EPGA process of designing/using dependency structures, mass calculation, present state factor (PSF), implementation state factor (ISF), transition state factor (TSF), performance score index, weighted performance score, and finally the ideal solution(s) determination are shown in the white boxes.
2. *Identify the goals; Define the criteria ranks (orange boxes):* The user should first start with performing a base cybersecurity assessment. Afterwards, they should define their goal in terms of desired cybersecurity posture and rank their criteria.
3. *Determine solution structure; remove duplicates (yellow boxes):* CyFEr will use built-in filters in combination with any user-defined filters to identify the most relevant solutions based on the goals. In the presented version, CyFEr has three filters. Upon acquiring the solutions from each filter, CyFEr then ensures that none of the solutions are repeated by running an internal process to remove duplicates.
4. *EPGA process (white boxes):* As part of the EPGA process, CyFEr uses the built-in dependency structures to calculate masses. The user has the ability to use their own dependency structures. Once the masses are calculated, CyFEr computes the PSF. During the ISF calculation, the user should choose from soft or hard methods. Based on the choice, CyFEr will perform soft or hard TSF. Note that the purple arrows associated with EISF and ETSF indicate that the user can perform those analysis either through soft or hard. The tool does not let user to mix soft and hard. For example, if ESISF is chosen, the tool automatically performs ESTSF as the next step. Finally, PSI and WPS are performed to find the ideal solution(s) which concludes EPGA and CyFEr.

3.1. PART – 1: DEPENDENCY STRUCTURES

A logical relationship can be defined across a set of complex cybersecurity controls through dependency structures. To design the dependency structures, concepts of "Set Theory" [61] are used where each subdomain (or criteria) is treated as a "set." In this scenario, the "objects of the set" are the "controls of the subdomain/criteria." Therefore, the mathematical construct is defined as follows:

Definition: The parent-child relationship between a set of security controls, C_i, in a criterion is irreflexive and asymmetric for those controls to form a logical, relative, and linear dependency structure.
Logical Construct: Let each subdomain be a set denoted as C_i, such that,

$$C_i \in \{C_1, C_2, ..., C_n\} \tag{1a}$$

Let dependency relationships of intra subdomain controls be denoted by R, such that

$$R \subset C_i \times C_j, S.T. \{C_i, C_j\} \in \{C_1, C_2, ..., C_n\} \tag{1b}$$

(1b) implies that R is a relation of C_i *and* C_j. An example of R may look like: R = $\{(C_1, C_2), (C_2, C_4), (C_3, C_6), \ldots\ldots\}$. Therefore, R is a finite set of ordered pairs of the form,(C_1, C_2), where, $\{C_1, C_2\} \in \{C_i, C_j\}$, where C_1 = Child Control; C_2 = Parent Control. Based on the above analysis, dependency relationship can be expressed as:

1. *Irreflexive*: For every element,$C_1 \in C_i$, $(C_1, C_1) \notin$ R [62]
2. *Asymmetric*: For all $C_1, C_2 \in C_i$, if $(C_1, C_2) \in$ R, then $(C_2, C_1) \notin$ R and $(C_1, C_1) \notin$ R [62].

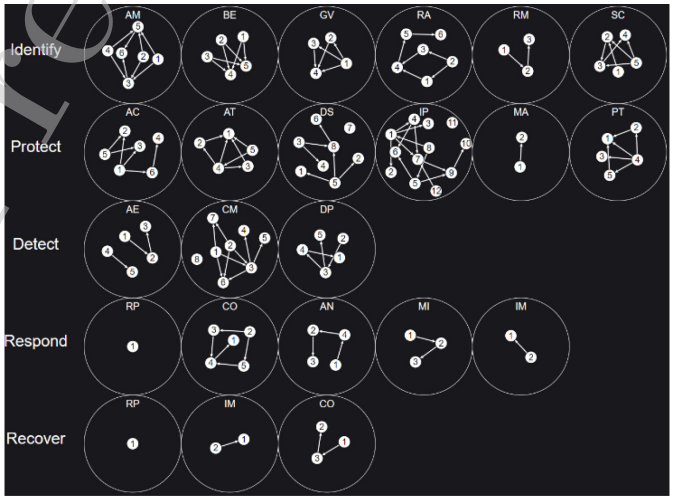


FIGURE 2. Dependency Structures of the Cybersecurity Framework

Based on the above definition and hypothesis, the relationships between the security controls of CSF are established pertaining to the below constrains:

1. MIL of a *parent* security control should be greater than or equal to the MIL of their respective *child* security controls. Therefore, MIL3 security controls are qualified to be parents to the security controls that belong to MIL3, MIL2, and MIL1. This relationship is maintained across the security controls that belong to all the MILs.

$$C_i^{MILa} \rightarrow C_j^{MILb}, S.T\ a \le b; \tag{1c}$$

$$\Rightarrow C_i^{MIL3} \rightarrow \begin{cases} C_j^{MIL3} \\ C_k^{MIL2}; C_i^{MIL2} \rightarrow \begin{cases} C_k^{MIL2} \\ C_m^{MIL1}; C_i^{MIL1} \rightarrow \{C_m^{MIL1} \end{cases} \\ C_m^{MIL1} \end{cases} \tag{1d}$$

"$\rightarrow$" indicated *parent to*

2. The relationship between *parent* controls and their *child* controls is based on one requirement—*child* controls are the absolute pre-requisites of the *parent* controls. The term "absolute pre-requisites" indicates that without mitigating the security vulnerabilities related to the *child* controls, the security vulnerabilities associated with the *parent* control cannot be mitigated.
3. MILs and stages are mutually exclusive (i.e., a control can reside at a stage regardless of its MIL and dependencies).

The result of the above process is shown in FIGURE 2, in which each numbered circle indicates the security control in the NIST CSF (see [14]).[1] Upon establishing the dependency trees across all CSF criteria, the relative masses of the security controls are calculated according to the formulation in *Part–2: Priority Stages and Masses.*

3.2. PART – 2: PRIORITY STAGES AND MASSES

After defining the dependency trees, the relative masses of the security controls are calculated based on one of the three methods shown in this section. To calculate the relative masses, a relative base mass that is valid for all the controls is defined. In conjunction with the already defined scale of MILs (i.e., MIL1, 2, and 3), the relative base masses (denoted as *base*) are defined as below:

$$C_{i|MIL1}^{base} = 0.1; C_{i|MIL2}^{base} = 0.2; C_{i|MIL3}^{base} = 0.3; \quad (2)$$

The base masses, MIL value, and stage of the control are used to calculate the relative mass of the control through one of the below methods:[2]

1. *Method-1:* The relative *mass* of a control, C_i, is C_i^{mass}; this can be obtained by the summation of masses of the dependencies and the base mass of the control, C_i:

$$C_i^{mass} = C_i^{base} + \sum_{k=1}^{n} C_k^{mass} \quad (3)$$

2. *Method-2:* The relative mass of a control, C_i, is C_i^{mass}; this can be obtained by the summation of masses of dependencies and the product of base mass and the stage-MIL factor of the control, C_i:

$$C_i^{mass} = \left(C_i^{base} \times C_i^{stage-MIL}\right) + \sum_{k=1}^{n} C_k^{mass} \quad (4a)$$

The stage-MIL factor of C_i is:

$$C_i^{stage-MIL} = C_i^{stage} + C_i^{MIL}$$

$$\Rightarrow C_i^{mass} = \left(C_i^{base} \times \left\{C_i^{stage} + C_i^{MIL}\right\}\right) + \sum_{k=1}^{n} C_k^{mass} \quad (4b)$$

where

$$C_k^{mass} = \left(C_k^{base} \times \left\{C_k^{stage} + C_k^{MIL}\right\}\right) \quad (4c)$$

3. *Method-3:* The relative *mass* of a control, C_i, is C_i^{mass}: this can be obtained by the summation of mass of the dependencies and the product of base mass and MIL of the control, C_i:

$$C_i^{mass} = \left(C_i^{base} \times C_i^{MIL}\right) + \sum_{k=1}^{n} C_k^{mass} \quad (5a)$$

where

$$C_k^{mass} = \left(C_k^{base} \times C_k^{MIL}\right) \quad (5b)$$

where
C_i^{mass} is the mass of the parent control Q_i,
C_i^{base} is the base mass of the parent control Q_i,
C_i^{stage} is the stage at which the parent control Q_i,
C_i^{MIL} is the MIL at which the parent control Q_i,
C_k^{mass} is the mass of the child control $k \in \{1, 2, \ldots, n\}$,
C_k^{base} is the base mass of the child control k,
C_k^{stage} is the stage of the child control k,
C_k^{MIL} is the MIL of the child control k.

Now that the relative mass values are computed for all the controls, the following section highlights a sequential list of steps that highlight EPGA formulation that will potentially result in an ideal maturity state of the system or a facility. In the above approach, note that the children (dependencies) and the respective parent controls will never be at the same stage.

3.3. PART – 3: EPGA FORMULATION

The core formulation of EPGA can be implemented and executed following a pre-defined sequence of steps that are coupled with the dependency structures and relative masses calculated in the previous sections. The following sequence of steps will demonstrate the core formulation of EPGA:

1) Step – 1: Identify the goal(s)

The first step in the analysis is to identify the goal(s) of the analysis. An example of the desired state can be as simple as "given the current state at 50% maturity (Note: Any control that is in state 3 or 4 [largely or fully implemented] is considered as a mature control), the final posture or desired state should be at 70% maturity."

2) Step – 2: Define the criteria ranks

In the presented framework, the weight of each criterion is determined using the rank-weight formulation defined in TABLE 1. In all of the rank-weight methods below, normalized relative weights were computed according to [63] [64].

Note that the summation of normalized weights of all criteria is always equal to 1. Therefore, (6a)–(6d) will always

[1] The arrow head in FIGURE 2 points to a parent control of its respective child(ren) control(s).

[2] Here, the term stage is defined as the total number of child layers lead to a parent control. For example, in the dependency structure $C_a \rightarrow C_b \rightarrow C_c$, the total number of stages associated with C_b are one and the total number of stages associated with C_c are two.

abide by the following constraint:

$$\sum_{i=1}^{n} W_{i|norm}^{\alpha} = 1, \alpha \in \{RS, RR, RE, ROC\} \quad (6e)$$

TABLE 1. Domains and Subdomains, Dependency trees based on the controls of NIST Cybersecurity Framework

	Method	Definition
1	Rank Sum (RS) [15]	For n number of criteria, the normalized weight $W_{i\|norm}^{RS}$ of criteria i of rank r is given by: $$W_{i\|norm}^{RS} = \frac{2(n-r+1)}{n(n+1)} \quad (6a)$$
2	Reciprocal Rank (RR) [15]	For n number of criteria, the normalized weight $W_{i\|norm}^{RR}$ of criteria i of rank r is given by: $$W_{i\|norm}^{RR} = \frac{(1/r_i)}{\sum_{j=1}^{n}(1/r_j)} \quad (6b)$$
3	Rank Exponent (RE) [25]	For n number of criteria, the normalized weight $W_{i\|norm}^{RE}$ of criteria i of rank r is given by: $$W_{i\|norm}^{RE} = \frac{(n-r_i+1)^p}{\sum_{j=1}^{n}(n-r_j+1)^p} \quad (6c)$$ $$\Rightarrow W_{i\|norm}^{RE} = \frac{\begin{cases}(n-r_i+1)^p, & \forall p \in (0,1) \wedge (1,\infty) \Rightarrow \forall p \in (0,\infty) \,\&\, p \neq 0,1,\infty \\ 1 & if\ p=0 \\ Rank\ Sum & if\ p=1\end{cases}}{\sum_{j=1}^{n}(n-r_j+1)^p}$$
4	Rank Order Centroid [68]	For n number of criteria, the normalized weight $W_{i\|norm}^{ROC}$ of criteria i of rank r is given by: $$W_{i\|norm}^{ROC} = \frac{1}{n}\sum_{j=i}^{n}\frac{1}{r_j} \quad (6d)$$

The normalized relative weights computed in (6) are used later in the paper to identify ideal solution(s) that will result in a desired cybersecurity posture.

$$RW_{net} = \begin{cases} \sum_{k=1}^{p} \text{Criteria}_{RW}^{k}; & if\ Criteria_{num} > 1 \\ 1; & if\ Criteria_{num} = 1 \end{cases} \quad (7a)$$

such that

$$RW_{net} = 1 \quad (7b)$$

Where

$\text{Criteria}_{\text{RW}}^{\text{k}}$ = RW value of criteria k; $0 \le k \le 1$

$\text{Criteria}_{\text{num}}$ = Total number of criteria

For a single criterion, RW value = RW_{net} = 1; for multiple criteria, the RW values should be defined for each criterion such that $\text{RW}_{\text{net}} = 1$ (see (7)).

According to the analysis and rank-weight method adaptions presented in [65] [66], it is evident that the above four methods are well-vetted and largely used in MCDA problems: 1) rank sum (RS), 2) reciprocal rank (RR), 3) rank exponent (RE), and 4) rank order centroid (ROC). Therefore, we adapted these four methods for the proposed methodology. Our future work will entail exploring the adaption and integration of other potential rank-weight methods. A detailed comparative analysis across the rank-weight methods shown in TABLE 1 was presented in [67]. For the EPGA adaption, the rank-weight methods' core formulations (shown in 6a – 6d) are used as is and without any modification. In our future work, we may investigate the need of any necessary modifications to the below rank-weight methods to further improve their adaption to the EPGA framework.

3) Step – 3: Determine solution structure

This step is further divided into two parts. The first part focuses on developing the sequential structure (M_S) that encapsulates all the controls of the vulnerability assessment framework. For example, the NIST Cybersecurity Framework has five domains, 23 subdomains, and 106 controls. The sequential structure would look like (8a). In (8a), M_S includes five elements, with each element representing a domain-structure. Each element of those domain-structures represents a subdomain-structure. Each subdomain-structure is denoted as $\text{PS}_{\text{j}}\left(\text{mat}\left(\text{t}_{1|2|3}\text{m}_{1|2|3}\right)\right)$, in which PS_j represents a particular subdomain, S_j, in domain P; $\text{mat}\left(\text{t}_{1|2|3}\text{m}_{1|2|3}\right)$ represents an NxM structure (N: total stages exists in subdomain; M: total MILs exists in subdomain), in which each element represents the total number of controls that falls into a particular stage and that belongs to a particular MIL. For example, an element in (8a) that shows t_1m_1 would be the total number of controls that are MIL1 residing in the first stage for subdomain S_j in domain P. The second part of this step focuses on using possibility generation (permutations) techniques to discover all the converged solutions that attempt to reach the desired cybersecurity state (goal). Constrain-based filtration should be used to eliminate unwanted solutions. With a base state of 0% maturity, there are over ~2^{32} possible solutions. Through testing, the total number of constrains were narrowed down to three, and solution sets were generated for those three constrains. The algorithm that encapsulates all three constrains below is shown in TABLE 2. The algorithm in TABLE 2 loops through the sequential structure (M_S) to call each constrain as functions to discover the solution sets.

- *Constrain – 1 (Solution Set – 1):* Given a desired end state of *x%* maturity, all the possible solutions that show *x%* of desired maturity being acquired in subdomains (criteria) should be discovered and grouped in the ranked order and in the order of lowest MIL control (MIL1) to highest MIL control (MIL3) until the overall maturity is achieved at *x%*.
- *Constrain – 2 (Solution Set – 2):* Given a desired end state of *x%* maturity, all the possible solutions that indicate up to 100% of each MIL control being achieved at either a state – 3 or state – 4 should be discovered and

TABLE 2. Algorithm: Step – 3: Determine Solution Structure

01: form structural matrix M_S
02: for M_S
03: call function method1() $\rightarrow S_1$
04: call function method2() $\rightarrow S_2$
05: call function method3() $\rightarrow S_3$
06: end for
07: compute final solution set $\rightarrow S_{final} = \bigcup_{i=1}^{n} S_i, here\ n = 3$
08: function method1() { //function method1() start
09: new solution s appended to new final solution set $\rightarrow s \in S_1^{future}$
10: *if* new solution is not in current final solution set, $S.T \rightarrow s \notin S_1^{present}$
11: while current maturity ≤ desired maturity $\rightarrow M_{net} \leq x\%$
12: for a criterion in a prioritized ranked criteria order
for $C_i, \forall C_i \in \{C_{Total}^{Ranked}\}, where\ C_{Total}^{Ranked} = \{C_1, C_2, ..., C_i, ..., C_n\}; n = 23$
13: while current maturity ≤ desired maturity $\rightarrow M_i^{net} \leq x\%$
14: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL1}, Q_i^{MIL1} \rightarrow S_{\{3\|4\}}$
15: then, move mil2 controls to li or fi $\rightarrow \forall Q_n^{MIL2}, Q_i^{MIL2} \rightarrow S_{\{3\|4\}}$
16: then, move mil3 controls to li or fi $\rightarrow \forall Q_n^{MIL3}, Q_i^{MIL3} \rightarrow S_{\{3\|4\}}$
17: end while
18: criteria++ //go to next criteria
19: end for
20: end while
21: return final solution set from method1 $\rightarrow S_1$ } //function method1() end
22: function method2() { //function method2() start
23: new solution s appended to new final solution set $\rightarrow s \in S_2^{future}$
24: *if* new solution is not in current final solution set, $S.T \rightarrow s \notin S_2^{present}$
25: while current maturity ≤ desired maturity $\rightarrow M_{net} \leq x\%$
26: for a criterion in a prioritized ranked criteria order $\rightarrow$
for $C_i, \forall C_i \in \{C_{Total}^{Ranked}\}, where\ C_{Total}^{Ranked} = \{C_1, C_2, ..., C_i, ..., C_n\}; n = 23$
27: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL1}, Q_i^{MIL1} \rightarrow S_{\{3\|4\}}$
28: criteria++ //go to next criteria
29: end for
30: for a criterion in a prioritized ranked criteria order $\rightarrow$
for $C_i, \forall C_i \in \{C_{Total}^{Ranked}\}, where\ C_{Total}^{Ranked} = \{C_1, C_2, ..., C_i, ..., C_n\}; n = 23$
31: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL2}, Q_i^{MIL2} \rightarrow S_{\{3\|4\}}$
32: criteria++ //go to next criteria
33: end for
34: for a criterion in a prioritized ranked criteria order $\rightarrow$
for $C_i, \forall C_i \in \{C_{Total}^{Ranked}\}, where\ C_{Total}^{Ranked} = \{C_1, C_2, ..., C_i, ..., C_n\}; n = 23$
35: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL3}, Q_i^{MIL3} \rightarrow S_{\{3\|4\}}$
36: criteria++ //go to next criteria
37: end for
38: end while
39: return final solution set from method2 $\rightarrow S_2$ } //function method2() end
40: function method3() { //function method3() start
41: new solution s appended to new final solution set $\rightarrow s \in S_3^{future}$
42: *if* new solution is not in current final solution set, $S.T \rightarrow s \notin S_3^{present}$
43: while current maturity ≤ desired maturity $\rightarrow M_{net} \leq x\%$
44: for a criterion in a prioritized ranked criteria order
for $C_i, \forall C_i \in \{C_{Total}^{Ranked}\}, where\ C_{Total}^{Ranked} = \{C_1, C_2, ..., C_i, ..., C_n\}; n = 23$
45: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL1}, Q_i^{MIL1} \rightarrow S_{\{3\|4\}}$
46: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL2}, Q_i^{MIL2} \rightarrow S_{\{3\|4\}}$
47: move mil1 controls to li or fi $\rightarrow \forall Q_n^{MIL3}, Q_i^{MIL3} \rightarrow S_{\{3\|4\}}$
48: criteria++
49: end for
50: end while
51: return final solution set from method3 $\rightarrow S_3$ } //function method3() end

grouped in the order of ranked criteria until the overall maturity is *x%*.

duplicates, eliminate the duplicates, and create a final solution set, S_{final}. This final solution set, S_{final}, is the union of all the three solution sets, as indicated in equation (8b).

$$M_S = \left(\begin{pmatrix} IS_1(mat(t_{1|2|3}m_{1|2|3})) \\ IS_2(mat(t_{1|2|3}m_{1|2|3})) \\ IS_3(mat(t_{1|2|3}m_{1|2|3})) \\ IS_4(mat(t_{1|2|3}m_{1|2|3})) \\ IS_5(mat(t_{1|2|3}m_{1|2|3})) \\ IS_6(mat(t_{1|2|3}m_{1|2|3})) \end{pmatrix} \begin{pmatrix} PS_1(mat(t_{1|2|3}m_{1|2|3})) \\ PS_2(mat(t_{1|2|3}m_{1|2|3})) \\ PS_3(mat(t_{1|2|3}m_{1|2|3})) \\ PS_4(mat(t_{1|2|3}m_{1|2|3})) \\ PS_5(mat(t_{1|2|3}m_{1|2|3})) \\ PS_6(mat(t_{1|2|3}m_{1|2|3})) \end{pmatrix} \begin{pmatrix} DS_1(mat(t_{1|2|3}m_{1|2|3})) \\ DS_2(mat(t_{1|2|3}m_{1|2|3})) \\ DS_3(mat(t_{1|2|3}m_{1|2|3})) \end{pmatrix} \begin{pmatrix} RS_1(mat(t_{1|2|3}m_{1|2|3})) \\ RS_2(mat(t_{1|2|3}m_{1|2|3})) \\ RS_3(mat(t_{1|2|3}m_{1|2|3})) \\ RS_4(mat(t_{1|2|3}m_{1|2|3})) \\ RS_5(mat(t_{1|2|3}m_{1|2|3})) \end{pmatrix} \begin{pmatrix} ES_1(mat(t_{1|2|3}m_{1|2|3})) \\ ES_2(mat(t_{1|2|3}m_{1|2|3})) \\ ES_3(mat(t_{1|2|3}m_{1|2|3})) \end{pmatrix} \right)$$

$$PS_j(mat(t_{1|2|3}m_{1|2|3})) = \begin{pmatrix} t_1m_1 & t_1m_2 & t_1m_3 \\ t_2m_1 & t_2m_2 & t_2m_3 \\ t_3m_1 & t_3m_2 & t_3m_3 \end{pmatrix}; P \in \begin{Bmatrix} \text{I=IDENTIFY} \\ \text{P=PROTECT} \\ \text{D=DETECT} \\ \text{R=RESPOND} \\ \text{E=RECOVER} \end{Bmatrix}; S_j \Rightarrow \text{SUBDOMAIN;MIN}(j)=1; max(j) = \text{TOTAL SUBDOMAINS IN DOMAIN } P \quad (8a)$$

- *Constrain – 3 (Solution Set – 3):* Given a desired end state of *x%* maturity, all the possible solutions that indicate up to 100% of controls being achieved at either a state – 3 or state – 4 should be discovered and grouped in the order of ranked criteria and in the order of MIL (MIL1→MIL2→MIL3) per criteria until the overall maturity is *x%*.

$$S_{final} = \bigcup_{i=1}^{n} S_i, here\ n = 3 \quad (8b)$$

Once the three solution sets are discovered and grouped in their respective families, the algorithm will check for

4) Step – 4: Calculate the enhanced present state factor

Now that a desired and potential solution set is discovered and obtained (S_{final}), the enhanced present state factor (EPSF) will be computed for the baseline (current cybersecurity maturity or posture). EPSF of the controls, subdomains (criteria), domains, and overall baseline is calculated through the below set of equations. To start with, *EPSF* for a control, *i,* can be calculated as:

$$EPSF_i = C_i^{mass} \times C_i^{(P_{state})} \quad (9a)$$

Following the three methods (*M1, M2, M3*) of relative mass calculation, the above equation may be written as:

$$EPSF_i^{M1} = \left(C_i^{base} + \sum_{k=1}^{n} C_k^{mass} \right) \times C_i^{(P_{state})} \quad (9b)$$

$$EPSF_i^{M2} = \left(\left(C_i^{base} \times \left\{ C_i^{stage} + C_i^{MIL} \right\} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_i^{(P_{state})} \quad (9c)$$

$$EPSF_i^{M3} = \left(\left(C_i^{base} \times C_i^{MIL} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_i^{(P_{state})} \quad (9d)$$

Where

$mil \in \{1,2,3\}$; $state \in \{1,2,3,4\}$;

C_i^{mil} = Maturity Indicator Level of a control i;

$C_i^{(P_{state})}$ = Present (chosen) state of a control i

$EPSF_i$ = EPSF for the control i *and* and $i \in \{1,2,\ldots,n\}$

5) Step – 5: Enhanced Implementation state factor

This step focuses on the calculation of the enhanced implementation state factor (EISF) for each solution. EISF is divided into soft enhanced implementation state factor (SEISF) and hard enhanced implementation state factor (HEISF). For a security control, *i*, the EISF is the product of mass of the security control and its state. If a control is at 3rd or 4th state, that control is already at the final state; therefore, zero jumps and SEISF is equal to 0. This is shown in (10a). HEISF is the sum of relative mass value of each control multiplied by the current state of control of solution *s*. If a control is at 4th state (this is the main difference between SEISF and HEISF), that control is at the final state; therefore, zero jumps. Thus, HISF is equal to 0. These relationships are shown in (10).

$$\{\xi\}\, EISF_i^s = \begin{cases} C_i^{mass} \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \quad (10a)$$

where

$$\xi = \begin{cases} S, & for\ Soft \\ H, & for\ Hard \end{cases} \text{ and } \delta = \begin{cases} 3, & for\ \xi = S \\ 4, & for\ \xi = H \end{cases} \quad (10b)$$

Following the three methods (*M1, M2, M3*) of relative mass calculation, the above equation may be written as:

$$\{\xi\}\, EISF_i^{s|M1} = \begin{cases} \left(C_i^{base} + \sum_{k=1}^{n} C_k^{mass} \right) \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \quad (10c)$$

$$\{\xi\}\, EISF_i^{s|M2} = \begin{cases} \left(\left(C_i^{base} \times \left\{ C_i^{stage} + C_i^{MIL} \right\} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \quad (10d)$$

$$\{\xi\}\, EISF_i^{s|M3} = \begin{cases} \left(\left(C_i^{base} \times C_i^{MIL} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \quad (10e)$$

6) Step – 6: Calculate the enhanced transition state factor

Enhanced transition state factor (ETSF) indicates the quantitative effort to transition from a base state (or present state) to a new (or potentially desired) state. ETSF is divided into a soft enhanced transition state factor (SETSF) and hard enhanced transition state factor (HETSF). For a security control, *i*, the SETSF is calculated using SEISF; HETSF is calculated using HEISF.

$$\{\xi\}\, ETSF_i^s = \{\xi\}\, EISF_i^s - EPSF_i \quad (11a)$$

where

$$\xi = \begin{cases} S, & for\ Soft \\ H, & for\ Hard \end{cases} \text{ and } \delta = \begin{cases} 3, & for\ \xi = S \\ 4, & for\ \xi = H \end{cases} \quad (11b)$$

$$\{\xi\}\, ETSF_i^s = \left(\begin{cases} C_i^{mass} \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \right) - \left(C_i^{mass} \times C_i^{(P_{state})} \right) \quad (11c)$$

Following the three methods (*M1, M2, M3*) of relative mass calculation, the above equation may be written as:

$$\{\xi\}\, ETSF_i^{s|M1} = \left(\begin{cases} \left(C_i^{base} + \sum_{k=1}^{n} C_k^{mass} \right) \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \right) - \left(\left(C_i^{base} + \sum_{k=1}^{n} C_k^{mass} \right) \times C_i^{(P_{state})} \right) \quad (11d)$$

$$\{\xi\}\, ETSF_i^{s|M2} = \left(\begin{cases} \left(\left(C_i^{base} \times \left\{ C_i^{stage} + C_i^{MIL} \right\} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \right) - \left(\left(\left(C_i^{base} \times \left\{ C_i^{stage} + C_i^{MIL} \right\} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_i^{(P_{state})} \right) \quad (11e)$$

$$\{\xi\}\, ETSF_i^{s|M3} = \left(\begin{cases} \left(\left(C_i^{base} \times C_i^{MIL} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_{i|state}^s; & if\ C_i^{(P_{state})} < \delta \\ 0; & if\ C_i^{(P_{state})} \geq \delta \end{cases} \right) - \left(\left(\left(C_i^{base} \times C_i^{MIL} \right) + \sum_{k=1}^{n} C_k^{mass} \right) \times C_i^{(P_{state})} \right) \quad (11f)$$

7) Step – 7: Calculate Performance Score Index

This step uses the criteria rankings calculated in Step – 2. Once the criteria rates are determined, the performance score indices (PSI) for each solution, *s,* can be calculated. PSI is a relative value (i.e., the smallest PSI is always 1 and the largest PSI is the number of constrain-based solutions discovered). As an example, (8b) results in a total number of solutions that is equal to the length of S_{final}. The PSI for a criteria per solution is assigned relative to PSI for the same criteria across the remaining solutions. Therefore, the PSI calculation and its range can be depicted as shown in (12a). According to (12a), PSI is an integer, and its value for a particular TSF of solution, *s,* pertaining to a criterion, *c,*

holds a linear relative dependency. Thus, the PSI value is equal to the relative state of ETSF for solution, *s,* relating to criteria, *c*.

$$PSI(ETSF_c^s) \rightarrow \begin{cases} \exists ETSF_c^s \in \mathbf{Z} PSI(ETSF_c^s); & PSI \in \{1,...,len(S_{final})\} \\ PSI_{\max} = len(S_{final}); & (ETSF_c^s)_{high} \\ PSI_{\min} = 1; & (ETSF_c^s)_{low} \end{cases} \quad (12a)$$

Note that all the discovered solutions will have the same cybersecurity posture of *x%*, which was as defined in the "goal" or desired maturity. For a criterion, the PSIs of the discovered solutions are based on the transition factor (separately using hard factor and soft factor). Therefore, the PSI of a solution of a criterion (note that the criteria = subdomain) depends on the transition state factor of that particular solution for that criteria (subdomain). This relationship is shown in (12b):

$$PSI_{sub}^s \propto ETSF_{sub}^s \quad (12b)$$

Therefore, a higher ETSF means a higher PSI and a lower TSF means a lower PSI (e.g., for three solutions, if S_1 has highest ETSF and S_3 has lowest ETSF, the PSI of S_1 is 3, S_2 is 2, and S_3 is 1).

8) Step – 8: Calculate Weighted Performance Score

The weighted performance score (WPS) is a scalar factor that is calculated using the PSI and the rank-weights (RW) obtained in Step – 2. Therefore,

$$WPS_s = \sum_{i=1}^{n} RW_i \times PSI_s^i \quad (13)$$

where

WPS_s = WPS for possibility solution s
RW_i = RW of critetion i for solution s
PSI_s^i = PSI of critetion i for solution s

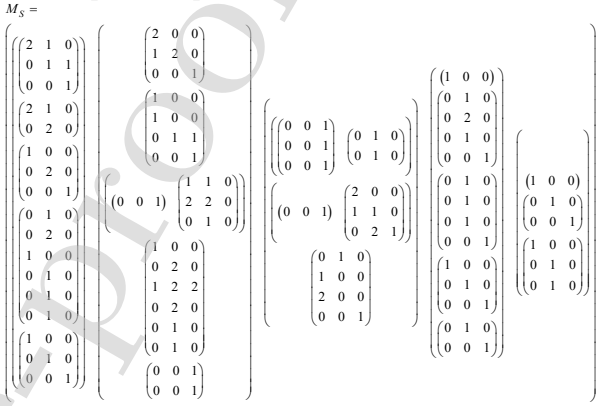
Upon calculating the WPS for all the solutions, a particular solution with the highest WPS value will be the ideal cybersecurity posture for the desired maturity of *x%*.

4. Real-World Attack-Based Experiment

A cyber-attack-based assessment was performed using the NIST Cybersecurity Framework (CSF) [14] to demonstrate the proposed methodology. Note that all the CSF controls used in this experiment can be seen and acquired from [14]. As shown in FIGURE 2, CSF has a total of 23 subdomains that are used as the criteria; those criteria are preferentially ranked. As mentioned in the previous sections, the dependency structures shown in FIGURE 2 are constructed by ingesting cybersecurity requirements defined in the NIST Cybersecurity Framework report and the MIL values associated with the controls in a criterion. If the dependency trees are changed, the algorithm is flexible and will be adapted to the new trees. Combining the dependency trees in FIGURE 2 with equation (8a) results in the sequential structure (M_S) that is shown in (14), which is used for the reminder of the calculations. The objectives of this experimental analysis are to complete the following:

1. Show that an attack such as the below could have been avoided had the facility used an assessment model like the NIST CSF, along with a prioritized mitigation framework, such as the EPGA, to evaluate their cybersecurity posture and mitigate the vulnerabilities that are critical to their mission and business continuity
2. Show the performance differences between EPGA Soft and EPGA Hard, and the respective repercussions
3. Demonstrate that the vulnerabilities identified by the EPGA for prioritized mitigation align with the vulnerabilities identified through subjective analyses shown in previous publications.

$M_S =$



(14)

4.1. Description of the Attack Scenario

According to the insider threat attack scenario demonstrated in [69] and as detailed in [70], [71], and [72], a third-party utility contractor carried out an insider attack on his ex-employer that caused over 200,000 gallons of sewage water to spill. Although the contractor had left his job due to disagreements with his employer, his credentials and access to the plant's critical control systems had not been revoked. Because he still had access to the critical systems, he was able to carry out a cyber-attack that produced a physical impact. A forensic analysis of the attack showed that the third-party contractor exploited weak SCADA systems and issued malicious commands while avoiding any monitoring or actionable alarms. The utility lacked basic cybersecurity systems, procedures, and policies that would prevent such an attack, including basic access controls defined in NIST SP 800-53 [73]. An independent manual assessment was completed to determine the required maturity to avoid or detect such an attack [69]. For the purpose of this experiment, both soft and hard PGA analysis was used to obtain a desired maturity without using the subjective methodology demonstrated in [69].

FIGURE 3 shows the criteria ranks and the computed relative weights based on the given ranks (note that the controllability parameter, *p,* for the RE method is assumed to be 2). The criteria are ranked through the performance of an attack analysis that was based on the deep-dive described in [69], [70], [71], and [72]. When criteria states are unchanged, they are ranked sequentially using the NIST domain and subdomain order.

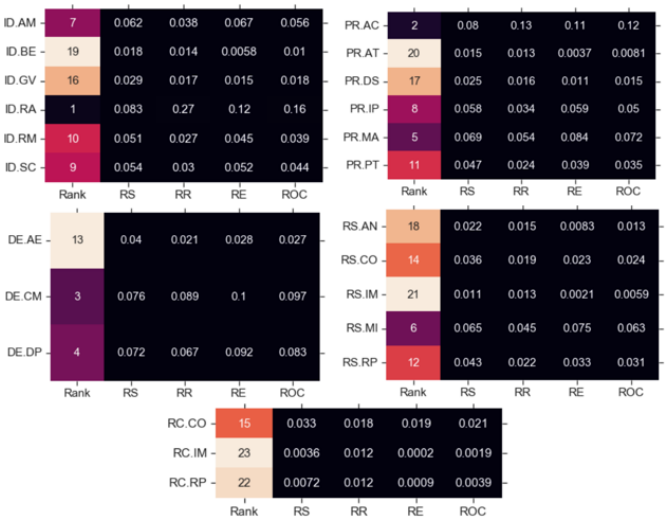


FIGURE 3. Criteria ranks, and calculated weights

4.2. EPGA Soft Analysis

According to the EPGA Soft formulation, the maturity of the facility that the attack was executed on was 57.5%; however, based on the subjective assessment performed in [69], the maturity should be at 87.7% according to the ranked order of criteria shown in FIGURE 3. This experiment was completed to determine the ideal solution that will result in a maturity of 87.7% using the EPGA Soft analysis.

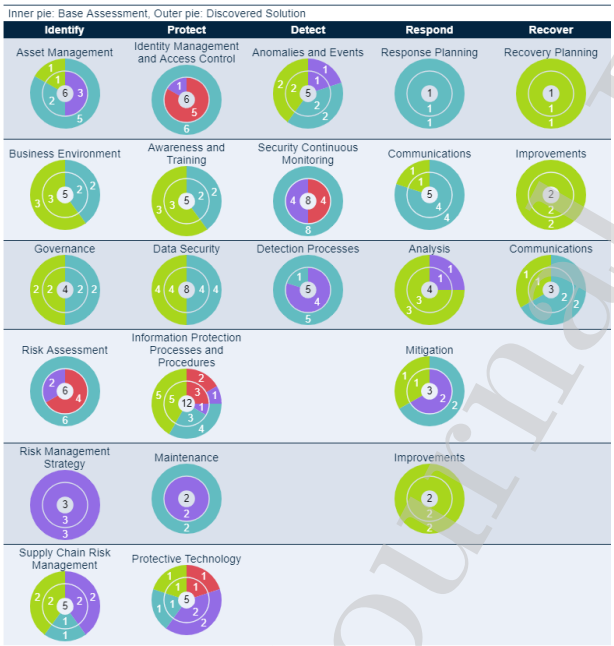


FIGURE 4. Domain-wise controls impacted in EPGA Soft solution

The results of the EPGA Soft analysis translated into a JavaScript-based web application are shown in FIGURE 4 and FIGURE 5. The highest WPS solution was analyzed, and detailed state changes of the controls were performed using the presented methodology. FIGURE 4 and FIGURE 5 show the total number of controls affected per criteria and per MIL discovered through analyzing further at the criteria level. As shown above, the EPGA Soft analysis considered both the criteria rank and the level of effort represented by the total number of controls expected to change states when determining the proposed solution with the desired maturity.

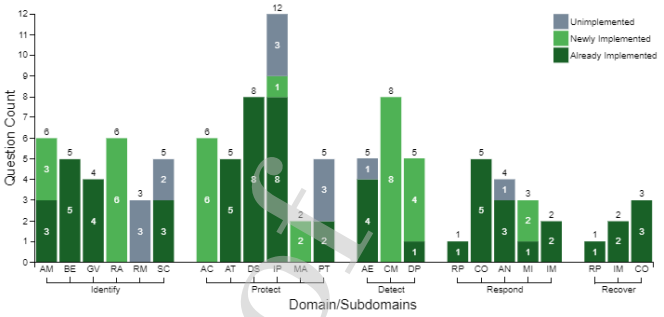


FIGURE 5. Depiction of impacted controls for EPGA Soft solution

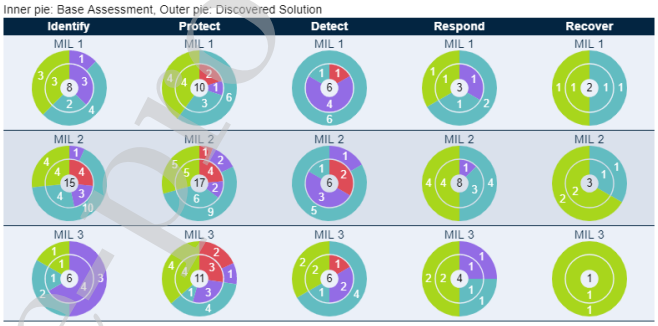


FIGURE 6. MIL-wise controls impacted in EPGA Soft solution

4.3. EPGA Hard Analysis

According to the EPGA Hard formulation, the maturity of the facility that the attack was executed on was 32.1%; however, based on the subjective assessment performed in [69], the maturity should be at 47.2% according to the ranked order of criteria shown in FIGURE 3. This experiment was completed to determine the ideal solution that will result in a maturity of 47.2% using the EPGA Hard analysis. The highest WPS solution was analyzed, and detailed state changes of the controls are shown in FIGURE 8 and FIGURE 9. It can be observed from FIGURE 9 that the security control transitions in the case of EPGA Hard are less than EPGA Soft (see FIGURE 5); however, the transitioned controls reached the *Fully Implemented* state.

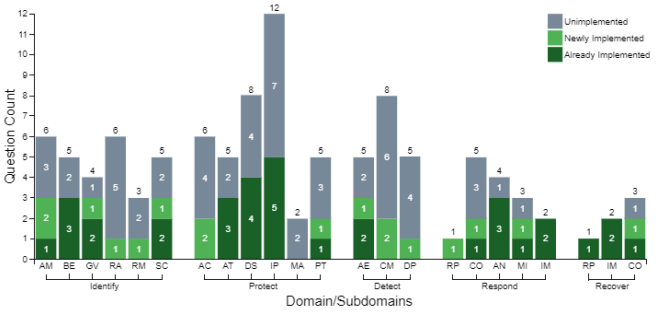


FIGURE 7. Depiction of impacted controls for EPGA Hard solution

Through further analysis at the criteria level, FIGURE 7 to FIGURE 9 show the controls affected per MIL and per criteria. As expected, the EPGA Hard analysis algorithm considered both the criteria rank and the level of effort

represented by the total number of controls expected to change states when finding a solution with the desired maturity.

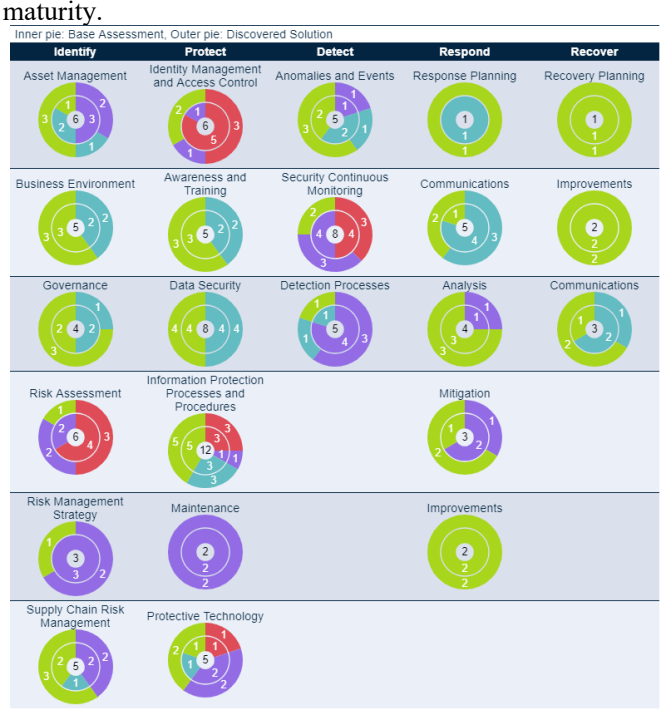


FIGURE 8. Domain-wise controls impacted in EPGA Hard solution

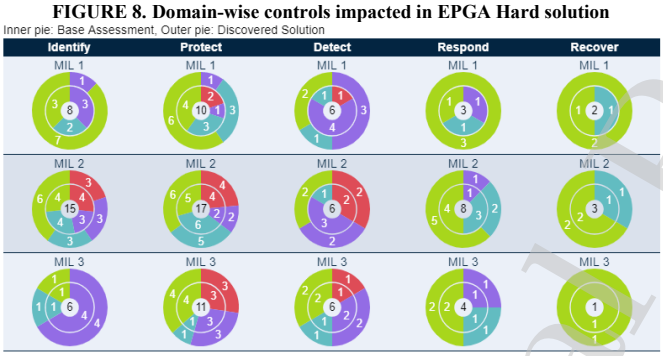


FIGURE 9. MIL-wise controls impacted in EPGA Hard solution

4.4. EPGA Soft vs Hard: Comparative Analysis

A comparative analysis was performed between the predecessor of the proposed EPGA solution, which is detailed in [74], the solution discovered through the EPGA Soft and EPGA Hard analysis, and the subjective solution described in [69]; the results are shown in FIGURE 10 and FIGURE 11. For contextual overview, EPGA's predecessor performs prioritized mitigation without accounting for dependency structures and relative mass values; the analysis is based on MIL values only. By setting that premise, note that the phrases *Base Soft* and *Sub Soft* indicate that for the base and subjective assessments, a control is fully mature if its state is either *Largely Implemented* or *Fully Implemented*. Similarly, the phrases *Base Hard* and *Sub Hard* indicate that for the base and subjective assessments, a control is fully mature if its state is *Fully Implemented*. FIGURE 10 and FIGURE 11 show that the overall maturity across the five domains is comparable between the five solutions (PGA Soft, EPGA Soft, PGA Hard, EPGA Hard, and subjective). When compared to the base maturity, it is clear that the EPGA analysis can provide a desired solution and eliminate a biased and rigorous process that a subjective decision-making analysis entails. FIGURE 10a shows that EPGA Soft analysis results focus on the *Detect* domain, with slightly less emphasis on the *Identify* and *Protect* domains. However, FIGURE 10b shows that the EPGA Hard analysis focused on the *Recover* domain, unlike PGA Hard targets (*Identify* and *Protect*). Comparing these outcomes with the dependency structure depicted in FIGURE 2, the above observations make logical sense; the EPGA algorithm approaches the solution of the given problem in a hierarchical fashion that is defined by the dependency trees. In such an approach, the EPGA algorithm gives its highest importance to the minimum required controls (also referred to as child controls). Therefore, domains with a highly controlled MIL distribution may gain a slightly higher importance. Note that the bias introduced through EPGA is very minimal and shows a promising balance in targeting the controls to transition them to a higher maturity. It is evident from FIGURE 10 that the EPGA result is better distributed and balanced across all domains when compared to PGA distribution. When comparing and analyzing the results shown in FIGURE 10c to FIGURE 10f at the MIL-level, EPGA behavior appears to be quite complex compared to PGA behavior. Behavioral inferences include the following. 1) EPGA Soft tends to be similar to PGA Soft for MIL1 controls, with a slightly higher emphasis on *Detect*, and EPGA Soft seems to target MIL3 controls more than MIL2 controls (which is quite opposite to PGA Soft behavior). 2) EPGA Hard clearly has a significant amount of affiliation to MIL1 controls, with about twice the emphasis on the *Response* and *Recover* domains as compared to PGA Hard. It can be noted that as the MIL increases, the EPGA solution tends to stay close to the PGA solution, with the exception of having a somewhat strong emphasis in the later domains (e.g., *Response* domain emphasis for MIL2 controls and *Detect* domains emphasis for MIL3 controls). EPGA appears to strictly follow the MIL progressive importance, distribution with fairly interesting domain-wise distribution patterns as compared to PGA (with the PGA seeming to be more concentrated in the beginning domains). Overall, it is clear that the subjective assessment/analysis tends to ignore that requirement, while the EPGA algorithm strictly follows the requirement pertaining to MIL progressiveness. Therefore, the solution discovered through the EPGA analysis is practical and achievable compared to the solution acquired through a subjective analysis. A detailed subdomain (criterion) level deep-dive is depicted in FIGURE 11 to compare, evaluate, and analyze the importance given to each criterion based on ranking and PSI. Based on the rankings shown in FIGURE 3, the top three criteria the PGA algorithm is expected to give the most importance to are *Identify – Risk Assessment, Protect – Identity Management and Access Control,* and *Detect – Security Continuous Monitoring*. It is evident from FIGURE 11a to FIGURE 11f

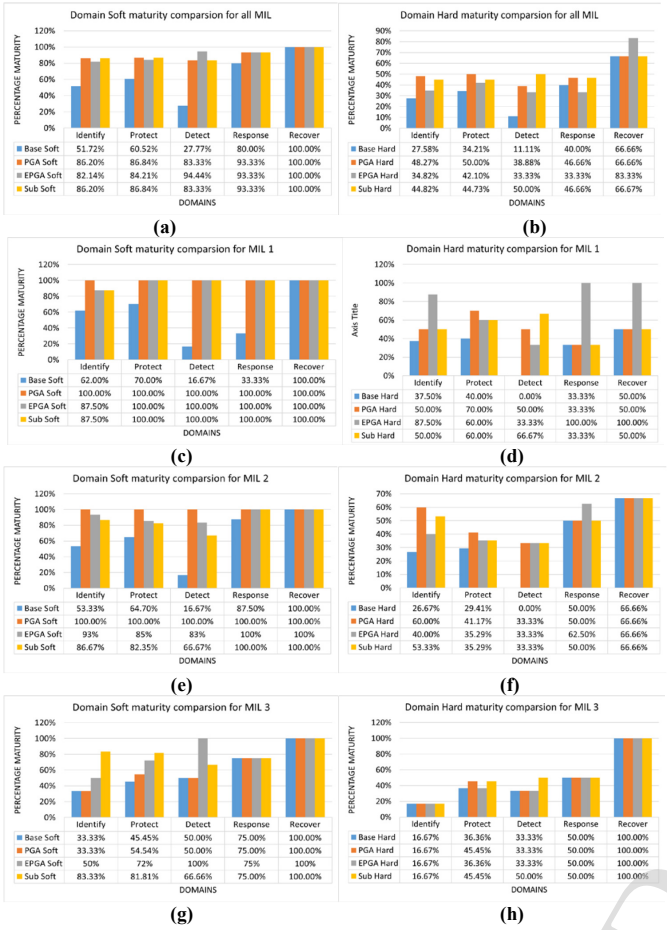


FIGURE 10. Comparative analysis of the subjective assessment, PGA Soft, EPGA Soft, PGA Hard, and EPGA Hard Analysis: (a) Overall comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the domains; (b) Overall comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the domains; (c) MIL1-based comparison between Base Soft, PGA Hard, EPGA Soft, and Subjective Soft across all the domains; (d) MIL1-based comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the domains; (e) MIL2-based comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the domains; (f) MIL2-based comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the domains; (g) MIL3-based comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the domains; (h) MIL3-based comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the domains.

that the criteria ranking is strictly respected; this is exclusively reflected in the EPGA Hard analysis (FIGURE 11b, FIGURE 11d, FIGURE 11f). The EPGA Soft analysis appears to be less aggressive and distributes state changes over several controls. Thus, the EPGA Soft algorithm's advantage is that it focuses on a wide range of criteria as well as reduces the risk of ignoring less important criteria. However, the EPGA Soft algorithm also has a drawback—it does not completely focus on forcing a criterion to reach the highest state of maturity. Similarly, the benefit of the EPGA Hard algorithm is that it focuses on a small range of criteria with a narrow and aggressive tactic to *force* the most important criteria to achieve the highest state of maturity. However, the EPGA Hard algorithm also has a drawback—it strictly prohibits the idea of distribution or dilution of importance across the entire range, which increases the risk of ignoring the less important criteria. The significant

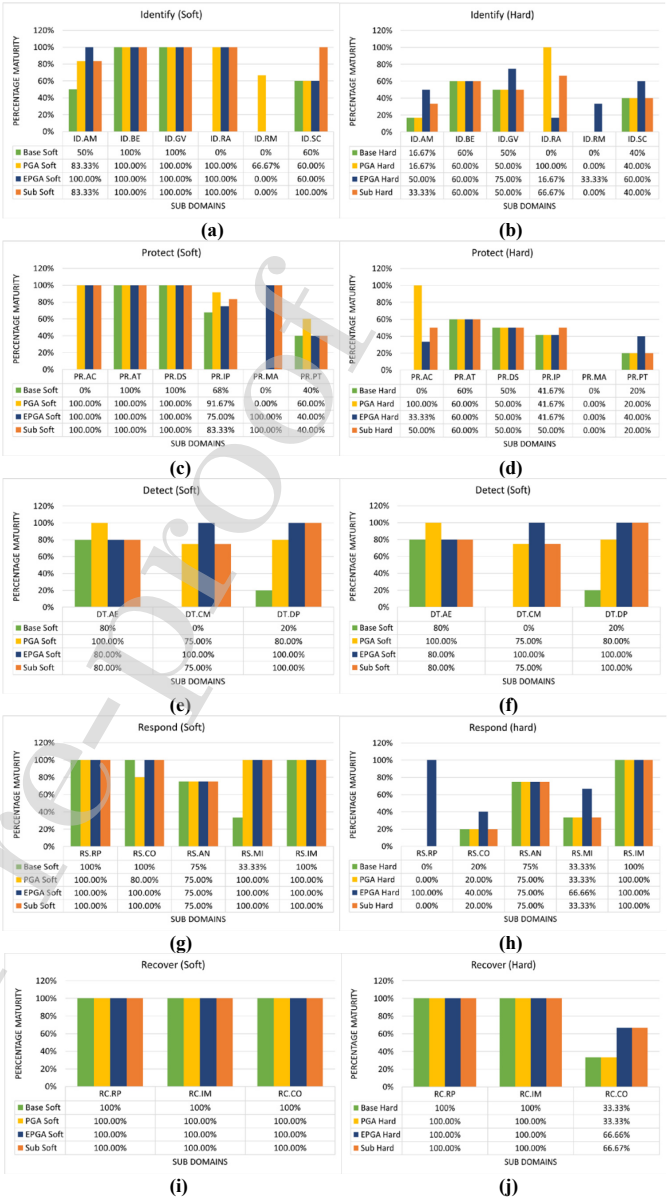


FIGURE 11. Domain-wise Comparative analysis of the base assessment, subjective assessment, PGA Soft, EPGA Soft, PGA Hard, and EPGA Hard Analysis: (a) Comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the subdomains in the Identify domain; (b) Comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the subdomains in the Identify domain; (c) Comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the subdomains in the Protect domain; (d) Comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the subdomains in the Protect domain; (e) Comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the subdomains in the Detect domain; (f) Comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the subdomains in the Detect domain; (g) Comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the subdomains in the Respond domain; (h) Comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the subdomains in the Respond domain; (i) Comparison between Base Soft, PGA Soft, EPGA Soft, and Subjective Soft across all the subdomains in the Recover domain; (j) Comparison between Base Hard, PGA Hard, EPGA Hard, and Subjective Hard across all the subdomains in the Recover domain.

inferences based on the obtained results are as follows: 1) The subjective solution lags significantly behind the EPGA Soft and EPGA Hard solutions. 2) The subjective solution included goals that may not be attainable due to the violation sequential domain-wise and criteria-wise

approaches and MIL progressiveness; the EPGA Hard and EPGA Soft solutions both respected those requirements. 3) Both the EPGA Hard and EPGA Soft analyses eliminate the whole process of a subjective analysis, but obtain the same overall domain-level maturity as the subjective analysis. Therefore, using a solution that was found using the EPGA Soft or EPGA Hard algorithm is achievable, objective, and supported by logical constructs—this removes the flaws of a subjective analysis, such as conflict of opinions, human-errors, and self-induced bias. 4) Since both the EPGA Hard and Soft algorithms are performed using a software program, the time needed to find and adopt a solution is much less than for a subjective approach, where everything must occur through a manual process and use a multi-party consensus. 5) An EPGA analysis removes the requirement of weighting criteria to reflect criticality and importance, which can be a roadblock (and often a requirement) in a subjective analysis. Overall, EPGA was found to be significantly more efficient than a subjective analysis.

Based on the detailed cyber-attack-based experimental analysis demonstrated in this section, it is evident that the objectives defined at the beginning of this section are clearly fulfilled. The EPGA Soft and Hard analyses shown in FIGURE 4 to FIGURE 9 identify the vulnerabilities that were needed to be mitigated to avoid the above attack. In addition, FIGURE 10 and FIGURE 11 show that EPGA Soft covers a broader range of security controls to achieve the minimum acceptable maturity state for the security control, whereas the EPGA Hard covers a small range of security controls to achieve the maximum acceptable maturity state for those identified security controls/vulnerabilities. FIGURE 10 and FIGURE 11 also show the performance of **the** EPGA Soft analysis perfectly aligns with the subjective analysis, whereas the EPGA Hard analysis attempts to reach that alignment, but falls slightly short due to the desired maturity limitations. Given the aggressive and selective nature of EPGA Hard, the behavior of the EPGA Hard analysis is expected.

5. Application and Comparative Analysis

Cybersecurity researchers developed various maturity models and assessment frameworks to evaluate the cybersecurity posture for different applications. The objective of this section is to provide a brief overview of the reputed cybersecurity maturity models and evaluate the application of EPGA to those models. In addition to such an application analysis, this section will also compare EPGA with some of the well-known models/approaches that are targeted to perform cybersecurity vulnerability analysis.

5.1. Applicability of EPGA to Various Maturity Model-based Vulnerability Assessments

Although this paper focuses on EPGA methodology and its application, we performed a study to evaluate the application of EPGA to other well-vetted and well-known maturity models. Based on the overview of various maturity models provided in [75] [76] [77] [78], we identified a total of 25 maturity models that have been in use across various industry sectors; TABLE 3 provides an overview of those models along with an evaluation against the application of EPGA to them. The first column of TABLE 3 lists the title of the maturity model, the second column lists the responsible entity (otherwise known as the entity that developed the respective maturity model), the third column provides an overview of the maturity model, the fourth column indicates if the maturity model security controls are categorized under a hierarchical fashion with maturity levels, and the fifth column estimates the potential of EPGA's application to the maturity model. Note that the fourth and fifth column follow a legend with the following delineations. 1) *Y-Y Category:* Y indicates yes. For 2–CSF, Y is indicated in the fourth and fifth columns because CSF has three maturity levels and EPGA is applied and tested with CSF. Other maturity models that fall under the Y-Y category are 1–C2M2, 3–FCF, and 4–SD2-C2M2. 2) *Y-YM Category:* Y indicates yes and YM indicates potentially yes, but requires testing. For 6–RMF, 7–NICE-CMM, 17–CPI, 19–SSE-CMM, 20–ISM3, 21–ISM2, and 24–COBIT, it is evident that the security controls are divided into multiple maturity levels. Therefore, EPGA has a very high potential to seamlessly integrate with those maturity models. However, testing is required to evaluate the efficacy of EPGA's application to those models. 3) *YC-YM Category:* YC indicates yes, but needs customization; the definition of YM remains the same as previously defined. For 5–CSET and 14–CRI, there are several annotations that can potentially be used to divide the security controls across multiple maturity levels. Upon achieving a successful categorization, EPGA can be applied. In such an integration, testing is required to determine if EPGA is fully compatible with those maturity models. 4) *N-N Category:* N indicates no. For 18–ISEM, 22–GISMM, and 23–ISF, N is indicated in the fourth and fifth columns because there is little to no information available on these models to estimate the applicability of EPGA. Therefore, these maturity models are defaulted under the N-N category. 5) *NC-YM Category:* NC indicates no, but can be potentially customized; the definition of YM remains the same as previously defined. For 8–CERT-RMM, 9–ISO/IEC 15408, 10–ISO/IEC 27001, 11–ISO/IEC 21827, 13–NCSecMM, 15–GCI, and 26–PCI-DSS, the controls were neither explicitly defined nor categorized under a multi-layer maturity level fashion. Therefore, EPGA can only be tested upon customizing these models to categorize under multiple maturity levels. 6) *NC-NM Category:* NM indicates that the possibility of EPGA's compatibility with the maturity model is lesser, but requires testing to confirm; the definition of NC remains same as previously defined. The only maturity model that falls under this category is 16–CMAPR. 7) *Y-NM Category:* The definitions of Y and NM remain the same as previously defined. The maturity model that falls under this category is 12–CCSMM; the guidelines in the maturity

model are divided across multiple maturity levels, but due to the amount of subjective analysis depicted, translating the guidelines to security controls is mandatory to evaluate the application of EPGA. In doing so, there may be some risk of minimal deviation from the objectives of the maturity model. Since no testing is performed with this category, no findings can be presented. 8) *YC-YM Category:* The only maturity model that falls under this category is 25–HIPAA because HIPAA is not a maturity model. However, there are multiple maturity models and assessment frameworks that are inherent parts of HIPAA. Therefore, some customization may be required to ensure that the security controls are categorized hierarchically under multiple maturity levels to test the applicability of EPGA. Upon successful categorization, EPGA integration should be potentially successful. Based on the above descriptive analysis, TABLE 3 is presented to provide a detailed comparative mapping overview of EPGA to the various cybersecurity assessment and maturity models.

5.2. Comparative Analysis

For decades, various cybersecurity vulnerability assessment and maturity models have been setting a precedent to enhance the security practices in a critical infrastructure facility. However, almost all of the well-known maturity models (see TABLE 3) are very subjective and are based on expert opinion. Lack of quantifiable methods to analyze the discovered vulnerabilities and the ability to prioritize those vulnerabilities to develop a mitigation strategy has been a challenge for the facility operators in translating the discoveries into actionable, affordable, and achievable best practices. As stated in [77], the efficacy of the maturity models is enhanced by coupling them with quantifiable frameworks to compute a numerical security level or to develop a mitigation approach supported by logical constructs. Similar to the presented work, there has been some work done by researchers to perform a quantifiable analysis in order to mitigate the discovered vulnerabilities.

As shown in TABLE 4, various quantification and vulnerability mitigation techniques are compared against the following eight areas:

1. *Approach:* Is the approach taken purely based on mathematical construct or adoptive/adaptive techniques such as machine learning/artificial intelligence? As shown in TABLE 4, the approach developed by [79] is the only approach found to be using machine learning. Excluding EPGA, about 25% of the approaches are mathematical and the remaining methods did not take any quantifiable approach. As stated in Section – I, the lack of open source and free-to-use cybersecurity data makes it very challenging to develop adaptive and adoptive techniques in order to develop quantitative maturity models and vulnerability mitigation frameworks.
2. *Potential to prioritize vulnerabilities:* Based on the evaluated models and frameworks shown in TABLE 4, the models developed by [80] and [81] were found to have potential to perform a prioritized vulnerability analysis. Prioritized vulnerability list is imperative to developing a successful mitigation plan, but it was found that the majority of the evaluated models do not make this a focus.
3. *Analysis style:* The third judging criteria is the analysis style. The majority of the maturity models are developed based on expert opinion. A similar trend is observed in the models that were attempting to perform some level of quantification. Except for the approaches defined by [21] [82] [79] [81], all the models were subjective and based on expert opinion. Note that the literature review did not find any model or framework that is completely objective. Designing a purely objective model may strongly depend on taking an adaptive/adoptive approach. However, due to the data availability limitations, there have not been extensive objective frameworks.
4. *Quantify security level:* This criterion evaluates if a framework or model quantifies security level based on the assessment. The models designed by [76] [82] [79] [81] estimate and quantify security levels. However, those inferences were largely coupled with subjective analyses.
5. *Compatibility with MCDA techniques:* Throughout the literature review, we did not find a mitigation framework that used MCDA techniques to perform ranked criteria-based vulnerability mitigation.
6. *Dependency Structures:* Although the evaluated models did not use dependency structures, the design and approach defined in [83] [84] [85] were flexible and compatible to incorporate dependency structures into them. However, further testing beyond implementation may be required to validate the above statement.
7. *Ingest maturity level controls:* Not all the models are designed to ingest maturity models with hierarchical security controls categorized under different maturity levels. Although the majority of the models seem to have included this factor in their design, the models designed by [79] [86] [87] [88] [85] may not be able to ingest hierarchical multi-tiered security controls.
8. *Versatility:* Overall, it is observed that all the models except [81] are designed to fit certain maturity model(s). However, the extensive model in [81] can potentially be used with several maturity models. Overall, similar to EPGA, the model in [81] seems to be versatile. However, [81] indicates the inability to handle dependency structures and MCDA techniques to perform ranked criteria-based vulnerability mitigation analysis.

Based on the applicability and comparative analysis demonstrated in the above two sections, it is evident that the proposed EPGA technique has several advantages over some of the proposed quantitative and mitigation frameworks. Although the applicability of EPGA to various maturity models shown in TABLE 3 must be proven based on testing and evaluation, the flexibility of EPGA indicates a seamless integration with the majority of those maturity models. Ongoing work and future publications will attempt to demonstrate an extensive study on the above stated subject.

TABLE 3. Evaluation of EPGA Applicability to Various Cybersecurity Maturity Models

Legend: Y = Yes; **N** = No; **YM** = Potentially Yes – Requires testing; **NM** = Potentially No – Requires testing; **YC** = Yes but needs customization; **NC** = No but can be potentially customized

Cybersecurity policies and Maturity Model		Responsible Entity	Overview of the maturity model	Hierarchical model?	Applicability potential for EPGA
1	C2M2 (derivations: ES-C2M2, B-C2M2, ONG-C2M2)	DOE, PNNL	Cybersecurity capability maturity model (C2M2) was developed by the U.S. Department of Energy (DOE) to enable an organization to evaluate the cybersecurity maturity of its systems. Other derivations of C2M2 such as Buildings-C2M2 (B-C2M2) was developed by the Pacific Northwest National Laboratory (PNNL). The security controls of C2M2 are grouped under 10 domains: Risk Management; Asset, Change, and Configuration Management; Identity and Access Management; Threat and Vulnerability Management; Situational Awareness; Information Sharing and Communications; Event and Incident Response, Continuity of Operations; Supply Chain and External Dependencies Management; Workforce Management; Cybersecurity Program Management.	Y	Y
2	CSF	NIST	The cybersecurity framework (CSF) was developed by National Institute of Standards and Technology (NIST). This provides a detailed set of security controls that focus on developing individual profiles for operators. The security controls of CSF are distributed across 5 domains: Identify; Protect; Detect; Respond; Recover.	Y	Y
3	FCF	PNNL	Facility Cybersecurity Framework (FCF) provides a set of voluntary, risk-based, standards and best practices to help facility owners and operators better manage cybersecurity risks.	Y	Y
4	SD2-C2M2	PNNL	The Secure Design and Development Cybersecurity Capability Maturity Model (SD2-C2M2) is a tool that can be used by hardware, software, and system developers and integrators to assess their design and development practices and procedures against a set of best-practice concepts to determine the maturity level of their processes. The tool breaks the design and development process down into seven major phases covering Background and Foundation; Design; Build; Test; integrate; Deploy; Lifecycle & End-of-life.	Y	Y
5	CSET	U.S. DHS	The cybersecurity evaluation tool (CSET) developed by the U.S. Department of Homeland Security (DHS) guides users through a step-by-step process to assess their control system and information technology network security practices against recognized industry standards.	YC	YM
6	RMF	NIST	The risk management framework (RMF) is a set of policies and standards that are recommended to help secure networked information systems.	Y	YM
7	NICE-CMM [19]	DHS	The National Initiative for Cybersecurity Education (NICE) developed a capability maturity model (CMM) to let organizations customize their cybersecurity workforce to obtain the right personnel to protect and defend their networked information systems from various cyber threats. NICE-CMM is defined across three areas: process and analytics; integrated governance; skilled practitioners; technology for workforce development.	Y	YM
8	CERT-RMM [89]	CERT/ SEI	The computer emergency response team (CERT) Software Engineering Institute (SEI) developed the resilience management model (RMM). This defines various set of practices that are required to manage and improve operational resilience, security, and business continuity.	NC	YM
9	ISO/IEC 15408 [90]	ISO	This was published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) established principles and concepts of the security evaluation for information technology systems.	NC	YM
10	ISO/IEC 27001 [91]	ISO	This defines a set of standards to help organizations to secure their information assets.	NC	YM
11	ISO/IEC 21827 [92]	ISO	With the goal of ensuring good security engineering, this describes the essential characteristics of the security engineering processes that must exist in an organization	NC	YM
12	CCSMM [93]	UT	The objective of the community cybersecurity maturity model (CCSMM) developed by Dr. White from University of Texas (UT), San Antonio is to determine the cybersecurity postures of organizations, communities, and nations.	Y	NM
13	NCSecMM [94]	USM	The national cybersecurity maturity model (NCSecMM) was developed at the University at Souissi, Morocco (USM). This defines a cybersecurity model for countries to perform a country-level assessment or evaluation. This package includes a framework, maturity model, role assignment, and maturity model	NC	YM
14	CRI [20]	HGS	The cyber readiness index (CRI) developed by Hathaway global strategies (HGS) was designed to assign CRI value at country-level across 7 elements: National Strategy; Incident Response; E-Crime and law enforcement; Information sharing; Investment in research and development (R&D); Diplomacy and trade; Defense and crisis response	YC	YM
15	GCI [95]	ITU	The global cybersecurity index (GCI) developed by the International Telecommunication Union (ITU) international agency to score countries according to their cybersecurity efforts. The focus areas of GCI are Legal measures; Technical measures; Organizational measures; Capacity building; Cooperation.	NC	YM
16	CMAPR [96]	ASPI	The cyber maturity in the Asia-Pacific region (CMAPR) was developed by the Australian Strategic Policy Institute (ASPI). This provides a cyberspace assessment of Asia-pacific countries across the following domains: governance and legislation; law enforcement; military capacity; policy involvement; business and social engagement in cyber policy and security issues.	NC	NM

Cybersecurity policies and Maturity Model		Responsible Entity	Overview of the maturity model	Hierarchical model?	Applicability potential for EPGA
17	CPI [21]	BAH	The cyber power index (CPI) developed by Booz Allen Hamilton (BAH) to assess the cyber power of 19 G-20 countries except the European Union. The weighted criteria approach was completely subjective and was defined by a panel of experts in the field.	Y	YM
18	ISEM [77]	CG	The information security evaluation maturity model (ISEM) was developed by City Group (CG) and categorized them security controls across the following domains: Complacency; Acknowledgement; Integration; Common Practice; Continuous Improvement. There is little to no information on this model.	N	N
19	SSE-CMM [97]	NSA	The systems security engineering capability maturity model (SSE-CMM) was developed by the U.S. National Security Agency (NSA) that has 5 maturity levels designed to evaluate/assess systems-level security practices.	Y	YM
20	ISM3 [22]	ISM3	The Information Security Management Maturity Model (ISM3) was developed by ISM3 consortium to evaluate, specify, implement, and enhance process-oriented information security management systems. The maturity levels used in this model are undefined; defined; managed; controlled; optimized.	Y	YM
21	ISM2 [23]	NIST-PRISMA	The information security maturity model (ISM2) was developed by NIST Program Review for Information Security Management Assistance (PRISMA) to review and measure the information security posture of an information security program. The security controls are spread across five domains: Policies; Procedures; Implemented; Tested; Integrated.	Y	YM
22	GISMM [77]	Gartner	The Gartner's Information Security Awareness Maturity Model designed by Gartner focuses on performing security awareness analysis and risk management in large organizations. There is very little information available on this model.	N	N
23	ISF [98]	IBM	The Information Security Framework (ISF) developed by IBM focuses on analyzing the security gaps between the business and underlying, overarching and related technologies. There is very little information available on this model.	N	N
24	COBIT [99]	ISACA	The Control Objectives for Information and Related Technology (COBIT) model (also referred to as Control Target Management Guidelines) was developed by Information Systems Audit and Control Association (ISACA) to perform information technology governance and management. This model has 5 maturity levels with the control spread across 4 domains: Planning and Organization; Delivering and Support; Acquiring and Implementation; Monitoring and Evaluating	Y	YM
25	HIPAA [100]	U.S. HSS	The Heath Insurance Portability and Accountability Act coined by the U.S. Department of Health and Human Services contains a security risk assessment tool and HIPAA Security Rule (HSR) toolkit that focuses on data privacy and security provisions for safeguarding medical information	YC	YM
26	PCI-DSS [101]	PCI	The Payment Card Industry Data Security Standard (PCI-DSS) was designed to ensure the secure design and implementation of the payment card and banking systems. The standards state various security controls to be assessed against and to be implanted to ensure cyber secure operations.	NC	YM

6. Conclusion

With the advancements in various technologies, such as IoT/IIoT, there has been a growing affinity towards integration of smart devices and networked systems across the global critical infrastructure landscape. Although such connected intelligent networks bear several benefits, such as precise data acquisition, decentralized automation and controls, and others, they are not inherently capable of protecting themselves from cyber-attacks. Therefore, it is the responsibility of the critical infrastructure owners and operators to ensure that their facility of networked systems is protected from and defended against cyber-attacks. In order to protect those systems, it is essential for the operators to understand their cybersecurity posture and a prioritized list of existing vulnerabilities that needs to be mitigated. To address the growing security needs, over the last few decades cybersecurity researchers and government agencies have developed several maturity model-based vulnerability assessments tools and frameworks. These tools and frameworks were designed to equip the critical infrastructure owners to assess their facilities and networks and understand the cybersecurity posture. A cybersecurity vulnerability assessment is an efficient way to determine the cybersecurity maturity and posture of a critical infrastructure facility. Several of the existing cybersecurity frameworks, models, and tools are designed to efficiently identify cybersecurity vulnerabilities in a facility, network, or system. However, several of those tools are qualitative and provide little to no means to develop plans and procedures to mitigate the discovered vulnerabilities. Realizing that need, researchers have been developing relative quantification-based risk assessment techniques and methodologies. Although these techniques can perform an effective risk analysis, they still lack the ability to precisely evaluate and ingest large-scale, system-level security controls to develop prioritized vulnerability mitigation strategies based on the desired cybersecurity maturity. The presented framework addresses that gap by prioritizing the vulnerabilities that need to be mitigated. Such a process will lead to a cybersecure path forward to defend and protect the critical infrastructure smart and networked systems against cyber intrusions.

TABLE 4. Comparison between EPGA and other quantitative/mitigation frameworks

Mitigation/Quantification Technique	Mathematical Approach (MA) vs Machine learning/Artificial Intelligence/Deep Learning (MAD)	Automated prioritized vulnerabilities	Subjective (S) vs Objective (O) vs Hybrid (H)	Quantify Security level	Compatibility with rank-weight MCDA techniques	Ability to incorporate security dependencies	Ability to ingest hierarchical multi-maturity security controls	Versatility to use across various maturity models (Largely = L; Partially = p)
Karabacak [76]	MA	-	S	X	-	-	X	P
Atoum [80]	-	X	S	-	-	-	X	P
CSIS [82]	MA	-	H	X	-	-	X	P
Cylance [79]	MAD	-	H	X	-	-	-	P
Nnolim [86]	-	-	S	-	-	-	-	P
Zuccato [83]	-	-	S	-	-	X	X	P
Janssen [87]	-	-	S	-	-	-	-	P
Von Solms [84]	-	-	S	-	-	X	X	P
Ku [102]	-	-	S	-	-	-	X	P
Jo [81]	MA	X	H	X	-	-	X	L
Otoom [88]	-	-	S	-	-	-	-	P
Oracle® [85]	-	-	S	-	-	X	-	P
CPI [21]	MA	-	H	-	-	-	-	P
EPGA	***MA***	***X***	***H***	***X***	***X***	***X***	***X***	***L***

This paper presented a novel methodology and a framework—the EPGA—that can be employed along with some of the well-vetted vulnerability analysis frameworks (e.g., CSF, C2M2, and RMF) to not only perform cybersecurity vulnerability analysis, but also perform prioritized vulnerability mitigation analysis. In addition, this paper also demonstrated the effectiveness of EPGA by testing the framework on a real-world cyber-attack. Through the use of the attack-based analysis presented in this paper, it was indicated that the mitigation methodology called EPGA, which is an integral part of CyFEr, was able to efficiently evaluate several thousands of possible solutions and identify the ideal solution to reach a desired cybersecurity maturity. CyFEr determined the possible model solutions that could allow cybersecurity maturity to be met while ensuring all the user requirements were achieved. In this paper, we also showed the potential of EPGA (and CyFEr) to integrate with maturity models and frameworks beyond CSF and C2M2. Through this comparative analysis, it was evident that the proposed methods address several gaps in the existing research, tools, and literature. The focus of future work and publications will be on the enhancement of the mitigation methodology presented in this paper; enhancements will be needed to add a stage-based approach as well as necessary complexity to ensure that the controls are layered based on both their MILs and their relative weights/masses.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Elsevier Journal of Future Generation Computer Systems,* vol. 29, no. 7, pp. 1645-1660, 2013.

[2] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A Survey," *Elsevier Journal of Computer Networks,* vol. 54, no. 15, pp. 2787-2805, 2010.

[3] Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)," Statista, 2019. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[4] Actiontec, "Smart Home Devices Expected to Experience Double-Digit Growth Through 2022," 2018. [Online]. Available: https://www.actiontec.com/wifi-market-research/smart-home-devices-expected-to-experience-double-digit-growth-through-2022/.

[5] MarketWatch, "Building Automation and Control Systems Market Size, Share, Report, Analysis, Trends & Forecast to 2026," 2019. [Online]. Available: https://www.marketwatch.com/press-release/building-automation-and-control-systems-market-size-share-report-analysis-trends-forecast-to-2026-2019-01-02.

[6] R. Weber, "Internet of Things - New Security and privacy challenges," *Elsevier Journal on Computer Law and Security Review,* vol. 26, no. 1, pp. 23-30, 2010.

[7] C. Ten, M. Govindarasu and C. Liu, "Cybersecurity for Critical infrastructures: Attack and Defense Modeling," *IEEE Transactions on Systems, MAN, and Cybernetics: Part A: Systems and Humans,* vol. 40, no. 4, pp. 853-865, 2010.

[8] C. Ten, C. Liu and M. Govindarasu, "Vulnerability Assessment of Cybersecurity for SCADA Systems Using Attack Trees," in *IEEE Power Engineering Society General Meeting*, USA, 2007.

[9] C. Ten, C. Liu and M. Govindarasu, "Vulnerability Assessment of Cybersecurity for SCADA Systems," *IEEE Transactions on Power Systems,* vol. 23, no. 4, pp. 1836-1846, 2008.

[10] M. Mylrea, S. Gourisetti, C. Larimer and C. Noonan, "Insider Threat Cybersecurity Framework Webtool & Methodology: Defending Against Complex Cyber-Physical Threats," in *WRIT*, USA, 2018.

[11] S. N. G. Gourisetti, M. Mylrea, E. Gervais and S. Bhadra, "Multi-Scenario Use Case based Demonstration of Buildings Cybersecurity Framework Webtool," in *IEEE Symposium on Computational Intelligence Applications in Smart Grid*, USA, 2017.

[12] M. Mylrea, S. N. Gourisetti and A. Nicholls, "An Introduction to Buildings Cybersecurity Framework (BCF)," in *IEEE Symposium on Computational Intelligence Applications in Smart Grid*, USA, 2017.

[13] C. Glantz, S. Somasundaram, M. Mylrea, R. Underhill and A. Nicholls, "Evaluating the Maturity of Cybersecurity Programs for

Building Control Systems," Technical Report, Pacific Northwest National Laboratory, 2016.

[14] NIST, "Framework for Improving Critical Infrastructure Cybersecurity V1.1," NIST, 2018.

[15] W. G. Stillwell, D. A. Seaver and W. Edwards, "A comparison of weight approximation techniques in multiattribute utility decision making," *Organizational Behavior and Human Performance,* pp. 62-77, 1981.

[16] H. Chi and V. Yu, "Ranking generalized fuzzy numbers based on centroid and rank index," *Elsevier Journal on Applied Soft Computing,* vol. 68, pp. 283-292, 2018.

[17] W. Edwards, "How to Use Multiattribute Utility Measurement for Social Decisionmaking," *IEEE Transactions on Systems, MAN, and Cybernetics,* vol. 7, no. 5, pp. 326-340, 1977.

[18] M. Barfod and S. Leleur, Multi-criteria decision analysis for use in transport decision making:, Denmark: DTU Lyngby: Technical University of Denmark, 2014.

[19] Department of Homeland Security, "Cybersecurity Capability Maturity Model White Paper," U.S. DHS, 2014.

[20] M. Hathaway, C. Demchak, J. Kerben, J. McArdle and F. Spidalieri, "Cyber Readiness Index 2.0," Arlington, VA, 2015.

[21] Booz Allen Hamilton, "Cyber Power Index: Findings and Methodology," The Economist: Economist Intelligence Unit, 2011.

[22] ISM3 Consortium, "Information Security Management Maturity Model," ISM3, 2009.

[23] NIST, "Program Review for Information Security Assistance," NIST, 2017. [Online]. Available: https://csrc.nist.gov/Projects/Program-Review-for-Information-Security-Assistance/Security-Maturity-Levels.

[24] R. Roberts and P. Goodwin, "Weight Approximations in Multi-attribute Decision Models," *Journal of Multi-Criteria Decision Analysis,* vol. 11, pp. 291-303, 2002.

[25] GITTA, "Geographical Information Technology Training Alliance," Nov 2013. [Online]. Available: http://www.gitta.info/Suitability/en/html/Normalisatio_learningObject1.html. [Accessed Jan 2018].

[26] E. Roszkowska, "Rank Ordering Criteria Weighting Methods - A Comparative Overview," Optimum Studia Ekonomiczne, 2013.

[27] A. Ganin, P. Quach, M. Panwar, Z. Collier, J. Keisler, D. Marchese and I. Linkov, "Multicriteria Decision Framework for Cybersecurity Risk Assessment and Management," *Wiley Journal on Risk Analysis,* 2017.

[28] J. Rossebo, R. Wolthuis, F. Fransen, G. Bjorkman and N. Medeiros, "An Enhanced Risk-Assessment Methodology for Smart Grids," *IEEE Computer Society,* 2017.

[29] K. Mesker, "Adapting NIST Cybersecurity Framework for Risk Assessment," Chevron, 2014.

[30] M. Ramadlan, "Introduction and implementation OWSAP Risk Rating Management," OWASP, 2017.

[31] NERC, "SRI Enhancement NERC Performance Analysis Subcommittee," NERC, 2014.

[32] U.S. DOE and EPRI, "Integrating Electricity Subsector Failure Scenarios into a Risk Assessment Methodology," U.S. DOE, EPRI, 2013.

[33] J. Rossebo, F. Fransen and E. Luiijf, "Security for Smart Electricity Grids: Including Threat Actor Capability and Motivation in Risk Assessment for Smart Grids," 2016.

[34] O. Gadyatskaya, R. Jhawar, P. Kordy, K. Lounis, S. Mauw and R. Trujillo-Rasua, "Attack Trees for Practical Security Assessment: Ranking of Attack Scenarios with ADTool 2.0," in *International Conference on Quantitative Evaluation of Systems*, Glasgow, UK, 2016.

[35] R. Lundberg and H. Willis, "Deliberative Risk Ranking to Inform Homeland Security Strategic Planning," *Journal of Homeland Security and Emergency Management,* vol. 13, no. 1, pp. 3-33, 2016.

[36] S. Lichtenstein, P. Slovic, B. Fischhoff and M. Layman, "Judged Frequency of Lethal Events," *Journal of Experimental Psychology: Human Learning and Memory,* vol. 4, no. 6, p. 551, 1978.

[37] D. Kahneman and A. Tversky, "On the Study of Statistical Intuitions," *Cognition,* vol. 11, no. 2, pp. 123-141, 1982.

[38] P. Slovic, "Perception of Risk," *Science,* vol. 236, no. 4799, pp. 280-285, 1987.

[39] P. Slovic, M. Finucane, E. Peters and D. MacGregor, "Risk as Analysis and Risk as Feelings: Some thoughts about Affect, Reason, Risk, and Rationality," *Risk Analysis,* vol. 24, no. 2, pp. 311-322, 2004.

[40] W. Viscusi and R. Zeckhauser, "Recollection Bias and Its Underpinnings: Lessons from Terrorism-Risk Assessments," *HKS Faculty Research Working Paper Series RWP15-066,* 2015.

[41] L. Cox Jr, "Some limitations of "Risk=Threat×Vulnerability×Consequence," *Risk Analysis,* vol. 28, no. 6, pp. 1749-1761, 2008.

[42] L. Cox Jr, "Improving Risk-Based Decision Making for Terrorism Applications," *Risk Analysis,* vol. 29, no. 3, pp. 336-341, 2009.

[43] B. Ezell, S. Bennett, D. von Winterfeldt, J. Sokolowski and A. Collins, "Probabilistic Risk Analysis and Terrorism Risk," *Risk Analysis,* vol. 30, no. 4, pp. 575-589, 2010.

[44] G. Brown and L. Cox Jr, "How Probabilistic Risk Assessment Can Mislead Terrorism Risk Analysts," *Risk Analysis,* vol. 31, no. 2, pp. 196-204, 2011.

[45] K. Morgan, "Development and Evaluation of a Method for Risk Ranking," Doctoral Dissertation: Carnegie Mellon University, 1999.

[46] M. Morgan, B. Fischhoff, L. Lave and P. Fischbeck, "A Proposal for Ranking Risk within Federal Agencies," In: (Davies, J.C. ed.) Comparing Environmental Risks: Tools for Setting Government Priorities, Washington, DC: Routledge, 1996.

[47] M. Morgan, H. Florig, M. DeKay and P. Fischbeck, "Categorizing Risks for Risk Ranking," *Risk Analysis,* vol. 20, no. 1, pp. 49-58, 2000.

[48] K. Morgan, M. DeKay, P. Fischbeck, M. Morgan, B. Fischhoff and H. Florig, "A Deliberative Method for Ranking Risks (II): Evaluation of Validity and Agreement among Risk Managers," *Risk Analysis,* vol. 21, no. 5, pp. 923-923, 2001.

[49] K. Jenni, "Attributes for Risk Evaluation," Doctoral Dissertation: Carnegie Mellon University, 1997.

[50] H. Florig, M. Morgan, K. Morgan, K. Jenni, B. Fischhoff, P. Fischbeck and M. DeKay, "A Deliberative Method for Ranking Risks (I): Overview and Test Bed Development," *Risk Analysis,* vol. 21, no. 5, pp. 913-913, 2001.

[51] H. Willis, J. MacDonald Gibson, R. Shih, S. Geschwind, S. Olmstead, J. Hu, A. Curtright, G. Cecchine and M. Moore, "Prioritizing Environmental Health Risks in the UAE," *Risk Analysis,* vol. 30, no. 12, pp. 1842-1856, 2010.

[52] K. Scarfone and P. Mell, "An Analysis of CVSS Version 2 Vulnerability Scoring," in *International Symposium on Emperical Software Engineering and Measurement*, 2009.

[53] J. Wang, H. Wang, M. Guo and L. Zhou, "Ranking Attacks Based on Vulnerability Analysis," in *Hawaii International Conference on System Sciences*, Hawaii, 2010.

[54] J. Hong and A. Haqiq, "What Vulnerability Do We Need to Patch First?," in *IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014.

[55] H. Ghani, J. Luna and N. Suri, "Quantitative Assessment of Software Vulnerabilities Based on Economic-Driven Security Metrics," in *International Conference on Risks and Security of Internet and Systems*, 2013.

[56] M. Keramati, "New Vulnerability Scoring System for Dynamic Security Evaluation," in *International Symposium on Telecommunications*, 2016.

[57] P. Mell, K. Scarfone and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," NIST; Forum of Incident Response and Security Teams, 2007.

[58] M. Corporation, "Common Vulnerabilities and Exposures (CVE)," [Online]. Available: http://cve.mitre.org/.

[59] T. Saaty, The Analytic Hierarchy Process, McGraw-Hill, 1980.

[60] T. Saaty, Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process, Pittsburgh: RWS Publications, 1994.

[61] R. Stoll and H. Enderton, "Encyclopedia Britannica: Set Theory," [Online]. Available: https://www.britannica.com/science/set-theory. [Accessed 2017].

[62] X. Xiao, "Discrete Structures Recitation," [Online]. Available: http://people.cs.pitt.edu/~xiangxiao/cs0441/Recitation-7.pdf. [Accessed 2017].

[63] F. Barron and B. Barrett, "The efficacy of SMARTER - Simple Multi-Attribute Rating Technique Extended to Ranking," *Elsevier Acta Psychologica,* vol. 93, pp. 23-36, 1996.

[64] A. Filho, T. Clemente, D. Morais and A. Almedia, "Preference modeling experiments with surrogate weighting procedures for the PROMETHEE method," *European Hournal of Operational Research,* vol. 264, no. 2, pp. 453-461, 2018.

[65] F. H. Barron, "Selecting a best multiattribute alternative with partial information about attribute weights," *Acta Psychologica,* vol. 80, no. 1-3, pp. 91-103, 1992.

[66] E. ROSZKOWSKA, "Rank ordering criteria weighting methods – a comparative overview," *Optimum. Studia Ekonomiczne,* vol. 5, no. 65, pp. 14-33, 2013.

[67] B. Ahn, "Compatible weighting method with rank order centroid: Maximum entropy ordered weighted averaging approach," *Elsevier European Journal of Operational Research,* vol. 212, no. 3, pp. 552-559, 2011.

[68] S. Gourisetti, M. Mylrea and H. Patangia, "Application of Rank-Weight Methods to Blockchain Cybersecurity Vulnerability Assessment Framework," in *IEEE Annual Computing and Communication Workshop and Conference*, Las Vegas, NV, 2019.

[69] M. Mylrea, S. Gourisetti, C. Larimer and C. Noonan, "Insider Threat Cybersecurity Framework Webtool & Methodology: Defending Against Complex Cyber-Physical Threats," in *WRIT*, USA, 2018.

[70] M. D. Abrams, "Malicious Control System Cyber Security Attack Case Study - Maroochy Water Services, Australia," in *Annual Computer Security Applications Conference*, 2008.

[71] C. Blask, "ICS Cybersecurity: Water, Water Everywhere," Nov 2011. [Online]. Available: http://www.infosecisland.com/blogview/18281-ICS-Cybersecurity-Water-Water-Everywhere.html. [Accessed 2017].

[72] L. J. Van Leuven, "Water/Wastewater Infrastructure Security: Threats and Vulnerabilities," Handbook of Water and Wastewater Systems Protection, Springer Science + Business Media, LLC, 2011.

[73] NIST, "NIST 800-53 Security and Privacy Controls for Information Systems and Organizations," NIST, 2017.

[74] S. Gourisetti, M. Mylrea and H. Patangia, "Cybersecurity Vulnerability Mitigation Framework through Empirical Paradigm: Prioritized Gap Analysis," *IEEE Transactions,* 2019 (under review).

[75] W. Miron and K. Muita, "Cybersecurity Capability Maturity Models for Providers of Critical Infrastructure," Technology Innovation Management Review, 2014.

[76] B. Karabacak, S. O. Yildirim and N. Baykal, "A Vulnerability-driven cyber security maturity model for measuring national critical infrastructure protection preparedness," *International Journal of Critical Infrastructure Protection,* vol. 15, pp. 47-59, 2016.

[77] N. T. Le and D. B. Hoang, "Can maturity models support cyber security?," in *IEEE 35th International Performance Computing and Communications Conference*, Las Vegas, NV, 2016.

[78] G. Xiao-yan, Y. Yu-qing and L. Li-lei, "An Information Security Maturity Evaluation Mode," *Elsevier Procedia Engineering,* vol. 24, pp. 335-339, 2011.

[79] Cylance Consulting, "NIST Cybersecurity Framework Gap Analysis: Identify Security Weaknesses in your Critical Infrastructure," Cylance, Irvine, CA.

[80] I. Atoum, A. Otoom and A. A. Ali, "A holistic cyber security implementation framework," *Information Management and Computer Security Journal,* vol. 22, no. 3, pp. 251-264, 2014.

[81] K. Jo and D. Won, "Advanced information security management evaluation system," *KSII Transactions on Internet and Information Systems,* vol. 5, no. 6, pp. 1192-1213, 2011.

[82] CSIS, "Cyber security gap analysis," www.csis.dk.

[83] A. Zuccato, "Holistic security management framework applied in electronic commerce," *Elsevier Computers and Security Journal,* vol. 26, no. 3, pp. 256-265, 2007.

[84] R. Von Solms, K. Thomson and P. M. Maninjwa, "Information Security Governance control through comprehensive policy architectures," in *Information Security for South Africa*, Johannesburg, South Africa, 2011.

[85] Oracle, "Information Security:AConceptual Architecture Approach [White Paper]," Oracle, 2011.

[86] A. L. Nnolim, "A Framework and Methodology for Information Security Management," Lawrence Technological University, Southfield, MI, 2007.

[87] M. Janssen and K. Hjort-Madsen, "Analyzing Enterprise Architecture in National Governments: The Cases of Denmark and the Netherlands," in *40th Annual Hawaii International Conference on System Sciences*, Waikoloa, HI, 2007.

[88] A. Otoom and I. Atoum, "An implementation framework (IF) for the national information assurance and cyber security strategy (NIACSS) of Jordan," *The International Arab Journal of Information Technology,* vol. 10, no. 4, 2013.

[89] CERT SEI, "CERT Resilience Management Model Version 1.2," CERT Program, 2016.

[90] ISO, "ISO/IEC 15408-1:2009: Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model," International Organization for Standardization, 2009. [Online]. Available: https://www.iso.org/standard/50341.html.

[91] ISO, "ISO/IEC 27000 family - Information security management systems," International Organization for Standardization, [Online]. Available: https://www.iso.org/isoiec-27001-information-security.html.

[92] ISO, "ISO/IEC 21827:2008: Information technology -- Security techniques -- Systems Security Engineering -- Capability Maturity Model® (SSE-CMM®)," International Organization for Standardization, [Online]. Available: https://www.iso.org/standard/44716.html.

[93] G. White, "The community cyber security maturity model," in *IEEE International Conference on Technologies for Homeland Security*, 2011.

[94] M. El Kettani and T. Debbagh, "NCSecMM: A national cyber security maturity model for an interoperable national cyber security framework," in *European Conference on Conference on e-Government*, 2009.

[95] ABI Research, "Global Cybersecurity Index: Conceptual Framework," ITU, United Kingdom, 2014.

[96] T. Feakin, J. Woodall and K. Aiken, "Cyber Maturity in the Asia-Pacific Region," Australian Strategic Policy Institute, Barton, Australia, 2014.

[97] SSE-CMM, "Systems Security Engineering - Capability Maturity Model," 2007. [Online]. Available: http://www.ssecmm.org/index.html.

[98] IBM, "IBM Information Security Framework".

[99] Simplilearn, "What is COBIT? - Significance and Framework," [Online]. Available: https://www.simplilearn.com/what-is-cobit-significance-and-framework-rar309-article.

[100] U.S. HHS, "Health Information Privacy: The Security Rule," 2017. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/security/index.html.

[101] PCI Security Standards Council, "PCI DSS Quick Reference Guide: Understanding the Payment Card Industry Data Security Standard Version 2.0," 2010.

[102] C. Ku, Y. Chang and D. Yen, "National information security policy and its implementation: A case study in Taiwan," *Elsevier Telecommunications Policy Journal,* vol. 33, no. 7, pp. 371-384, 2009.

APPENDIX: EPGA DEMO THROUGH A PROBLEM

The objective of the below *toy example* is to demonstrate a walkthrough of the EPGA steps through a simple problem. Below are some important clarifications.

1. The demonstrated problem is extremely simplified with only six possible solutions. In reality, the problems tend to be complicated, with 100+ security controls. For example, C2M2 has 10 domains, over 30 subdomains (criteria), and over 350 controls. Similarly, CSF has five domains, 23 subdomains (criteria), and over 100 controls. Also, note that in reality, the web-tool mentioned earlier executes all the steps discussed.
2. Since the *toy example* below has a small number of controls, criteria, and domains, the effectiveness of the dependency structures is not fully reflected. These computations are performed manually in the example below to show the progress through each step.

In this problem, there are three criteria with five controls across them. According to TABLE 5, the current state of maturity is 20% and the desired maturity is 60%. The EPGA Soft analysis with M2 mass calculation is used to discover the ideal solution.

TABLE 5. Stages, MILs, masses, and base states of the illustrative security controls

Domain	Criteria	Control	Parent to	Stage	MIL	Mass (M2)	Base State
D1	C1	Q1	-	1	1	0.1 x (1+1) = 0.2	2
		Q2	Q1	2	3	(0.3 x (2+3)) +0.2 = 1.7	1
	C2	Q3	-	1	2	0.2 x (1+2) = 0.6	4
		Q4	Q3	2	3	(0.3 x (2+3)) +0.6 = 2.1	2
D2	C3	Q5	-	1	1	0.1 x (1+1) = 0.2	2

The current state can be calculated using:

$$Maturity = \frac{\sum_{i=1}^{n} Q_i ; S.T.\ \text{state of } Q_i = 3 \,||\, 4}{\sum_{i=1}^{m} Q_i ; for\ m\ controls} \times 100$$

$$\Rightarrow Maturity = \frac{(Q_3)}{5(Q_1 : Q_5)} \times 100 = 20\%$$

Step – 1: Identify the goal: Reach a desired maturity of 60%.

Step – 2: Define the criteria ranks (use (6)):

TABLE 6. Ranks and associated relative weights of the illustrative criteria

Criteria	Rank	W_i^{RS}	W_i^{RR}	W_i^{RE}	W_i^{ROC}
C1	2	0.333	0.2727	0.2857	0.2778
C2	3	0.167	0.1818	0.0714	0.1111
C3	1	0.5	0.5455	0.6429	0.6111

Step – 3: Determine Solution Structure: Since there are only five controls, potential solutions are easily discovered. In the case of CSF with a given base maturity and desired maturity, possible solutions may be in the range of at least ~2^{20} to ~2^{32} and potential solutions may be in the range of a few thousand.

TABLE 7. Discovered solution sets for the illustrative problem

Solution	Final States (Maturity = 60%)
S1	{3,3,4,2,2}
S2	{3,1,4,3,2}
S3	{3,1,4,2,3}
S4	{2,3,4,3,2}
S5	{2,3,4,2,3}
S6	{2,1,4,3,3}

Note that since the EPGA Soft analysis is performed, remaining solutions that result in a state change to "4" for either Q_1, Q_2, Q_4, or Q_5 are not considered. In TABLE 7, state changes of the controls across discovered solutions are shown in red.

Step – 4: Calculate Present State Factor (use (9)):

$PSF_{Q1} = 0.2 \times 2 = 0.4; PSF_{Q2} = 1.7 \times 1 = 1.7; PSF_{Q3} = 0.6 \times 4 = 2.4;$
$PSF_{Q4} = 2.1 \times 2 = 4.2; PSF_{Q5} = 0.2 \times 2 = 0.4$

$$\Rightarrow \begin{pmatrix} PSF_{C1} = PSF_{Q1} + PSF_{Q2} = 0.4 + 1.7 = 2.1; \\ PSF_{C2} = PSF_{Q3} + PSF_{Q4} = 2.4 + 4.2 = 6.6; \\ PSF_{C3} = PSF_{Q5} = 0.4 \end{pmatrix}$$

Step – 5a: Calculate SISF (use (10)):

TABLE 8. SISF calculation of the illustrative problem

S1	$SISF_{Q1}^{S1} = 0.2 \times 3 = 0.6; SISF_{Q2}^{S1} = 1.7 \times 3 = 5.1;$ $SISF_{Q3}^{S1} = 0.6 \times 4 = 2.4;\ SISF_{Q4}^{S1} = 2.1 \times 2 = 4.2;$ $SISF_{Q5}^{S1} = 0.2 \times 2 = 0.4$ $SISF_{C1}^{S1} = 0.6 + 5.1 = 5.7;\ SISF_{C2}^{S1} = 2.4 + 4.2 = 6.6;$ $SISF_{C3}^{S1} = 0.4$
S2	$SISF_{Q1}^{S1} = 0.2 \times 3 = 0.6; SISF_{Q2}^{S1} = 1.7 \times 1 = 1.7;$ $SISF_{Q3}^{S1} = 0.6 \times 4 = 2.4;\ SISF_{Q4}^{S1} = 2.1 \times 3 = 6.3;$ $SISF_{Q5}^{S1} = 0.2 \times 2 = 0.4$ $SISF_{C1}^{S1} = 0.6 + 1.7 = 2.3;\ SISF_{C2}^{S1} = 2.4 + 6.3 = 8.7;$ $SISF_{C3}^{S1} = 0.4$

S3	$SISF_{Q1}^{S1} = 0.2 \times 3 = 0.6; SISF_{Q2}^{S1} = 1.7 \times 1 = 1.7;$ $SISF_{Q3}^{S1} = 0.6 \times 4 = 2.4;\ SISF_{Q4}^{S1} = 2.1 \times 2 = 4.2;$ $SISF_{Q5}^{S1} = 0.2 \times 3 = 0.6$ $SISF_{C1}^{S1} = 0.6 + 1.7 = 2.3;\ SISF_{C2}^{S1} = 2.4 + 4.2 = 6.6;$ $SISF_{C3}^{S1} = 0.6$
S4	$SISF_{Q1}^{S1} = 0.2 \times 2 = 0.4; SISF_{Q2}^{S1} = 1.7 \times 3 = 5.1;$ $SISF_{Q3}^{S1} = 0.6 \times 4 = 2.4;\ SISF_{Q4}^{S1} = 2.1 \times 3 = 6.3;$ $SISF_{Q5}^{S1} = 0.2 \times 2 = 0.4$ $SISF_{C1}^{S1} = 0.4 + 5.1 = 5.5;\ SISF_{C2}^{S1} = 2.4 + 6.3 = 8.7;$ $SISF_{C3}^{S1} = 0.4$
S5	$SISF_{Q1}^{S1} = 0.2 \times 2 = 0.4; SISF_{Q2}^{S1} = 1.7 \times 3 = 5.1;$ $SISF_{Q3}^{S1} = 0.6 \times 4 = 2.4;\ SISF_{Q4}^{S1} = 2.1 \times 2 = 4.2;$ $SISF_{Q5}^{S1} = 0.2 \times 3 = 0.6$ $SISF_{C1}^{S1} = 0.4 + 5.1 = 5.5;\ SISF_{C2}^{S1} = 2.4 + 4.2 = 6.6;$ $SISF_{C3}^{S1} = 0.6$
S6	$SISF_{Q1}^{S1} = 0.2 \times 2 = 0.4; SISF_{Q2}^{S1} = 1.7 \times 1 = 1.7;$ $SISF_{Q3}^{S1} = 0.6 \times 4 = 2.4;\ SISF_{Q4}^{S1} = 2.1 \times 3 = 6.3;$ $SISF_{Q5}^{S1} = 0.2 \times 3 = 0.6$ $SISF_{C1}^{S1} = 0.4 + 1.7 = 2.1;\ SISF_{C2}^{S1} = 2.4 + 6.3 = 8.7;$ $SISF_{C3}^{S1} = 0.6$

Step – 6a: Calculate STSF (use (11)):

TABLE 9. STSF calculation of the illustrative problem

S1	$STSF_{C1}^{S1} = SISF_{C1}^{S1} - PSF_{C1} = 5.7 - 2.1 = 3.6$ $STSF_{C2}^{S1} = SISF_{C2}^{S1} - PSF_{C2} = 6.6 - 6.6 = 0$ $STSF_{C3}^{S1} = SISF_{C3}^{S1} - PSF_{C3} = 0.4 - 0.4 = 0$
S2	$STSF_{C1}^{S2} = SISF_{C1}^{S2} - PSF_{C1} = 2.3 - 2.1 = 0.2$ $STSF_{C2}^{S2} = SISF_{C2}^{S2} - PSF_{C2} = 8.7 - 6.6 = 2.1$ $STSF_{C3}^{S2} = SISF_{C3}^{S2} - PSF_{C3} = 0.4 - 0.4 = 0$
S3	$STSF_{C1}^{S3} = SISF_{C1}^{S3} - PSF_{C1} = 2.3 - 2.1 = 0.2$ $STSF_{C2}^{S3} = SISF_{C2}^{S3} - PSF_{C2} = 6.6 - 6.6 = 0$ $STSF_{C3}^{S3} = SISF_{C3}^{S3} - PSF_{C3} = 0.6 - 0.4 = 0.2$
S4	$STSF_{C1}^{S4} = SISF_{C1}^{S4} - PSF_{C1} = 5.5 - 2.1 = 3.4$ $STSF_{C2}^{S4} = SISF_{C2}^{S4} - PSF_{C2} = 8.7 - 6.6 = 2.1$ $STSF_{C3}^{S4} = SISF_{C3}^{S4} - PSF_{C3} = 0.4 - 0.4 = 0$
S5	$STSF_{C1}^{S5} = SISF_{C1}^{S5} - PSF_{C1} = 5.5 - 2.1 = 3.4$ $STSF_{C2}^{S5} = SISF_{C2}^{S5} - PSF_{C2} = 6.6 - 6.6 = 0$ $STSF_{C3}^{S5} = SISF_{C3}^{S5} - PSF_{C3} = 0.6 - 0.4 = 0.2$
S6	$STSF_{C1}^{S6} = SISF_{C1}^{S6} - PSF_{C1} = 2.1 - 2.1 = 0$ $STSF_{C2}^{S6} = SISF_{C2}^{S6} - PSF_{C2} = 8.7 - 6.6 = 2.1$ $STSF_{C3}^{S6} = SISF_{C3}^{S6} - PSF_{C3} = 0.6 - 0.4 = 0.2$

Step – 7: Calculate Performance Score Index (use (12)):

TABLE 10. PSI calculation of the illustrative problem

Criteria	S1	S2	S3	S4	S5	S6
C1	6	2	2	4	4	1
C2	1	4	1	4	1	4
C3	1	1	4	1	4	4

Step – 8: Calculate Weighted Performance Score (use (13)):

TABLE 11. WPS calculation of the illustrative problem

RANK SUM METHOD	
S1	$(6 \times 0.3333) + (1 \times 0.167) + (1 \times 0.5) = 2.6668$
S2	$(2 \times 0.3333) + (4 \times 0.167) + (1 \times 0.5) = 1.8346$
S3	$(2 \times 0.3333) + (1 \times 0.167) + (4 \times 0.5) = 2.8336$
S4	$(4 \times 0.3333) + (4 \times 0.167) + (1 \times 0.5) = 2.5012$
S5	$(4 \times 0.3333) + (1 \times 0.167) + (4 \times 0.5) = 3.5002$
S6	$(1 \times 0.3333) + (4 \times 0.167) + (4 \times 0.5) = 3.0013$
RECIPROCAL RANK METHOD	
S1	$(6 \times 0.2727) + (1 \times 0.1818) + (1 \times 0.5455) = 2.3635$
S2	$(2 \times 0.2727) + (4 \times 0.1818) + (1 \times 0.5455) = 1.8181$
S3	$(2 \times 0.2727) + (1 \times 0.1818) + (4 \times 0.5455) = 2.9092$
S4	$(4 \times 0.2727) + (4 \times 0.1818) + (1 \times 0.5455) = 2.3635$
S5	$(4 \times 0.2727) + (1 \times 0.1818) + (4 \times 0.5455) = 3.4546$
S6	$(1 \times 0.2727) + (4 \times 0.1818) + (4 \times 0.5455) = 3.1819$
RANK EXPONENT METHOD	
S1	$(6 \times 0.2857) + (1 \times 0.0714) + (1 \times 0.6429) = 2.4285$
S2	$(2 \times 0.2857) + (4 \times 0.0714) + (1 \times 0.6429) = 1.4999$
S3	$(2 \times 0.2857) + (1 \times 0.0714) + (4 \times 0.6429) = 3.2144$
S4	$(4 \times 0.2857) + (4 \times 0.0714) + (1 \times 0.6429) = 2.0713$
S5	$(4 \times 0.2857) + (1 \times 0.0714) + (4 \times 0.6429) = 3.7858$
S6	$(1 \times 0.2857) + (4 \times 0.0714) + (4 \times 0.6429) = 3.1429$
RANK ORDER CENTROID METHOD	
S1	$(6 \times 0.2778) + (1 \times 0.1111) + (1 \times 0.6111) = 2.2389$
S2	$(2 \times 0.2778) + (4 \times 0.1111) + (1 \times 0.6111) = 1.6111$
S3	$(2 \times 0.2778) + (1 \times 0.1111) + (4 \times 0.6111) = 3.1111$
S4	$(4 \times 0.2778) + (4 \times 0.1111) + (1 \times 0.6111) = 2.1667$
S5	$(4 \times 0.2778) + (1 \times 0.1111) + (4 \times 0.6111) = 3.6667$
S6	$(1 \times 0.2778) + (4 \times 0.1111) + (4 \times 0.6111) = 3.1666$

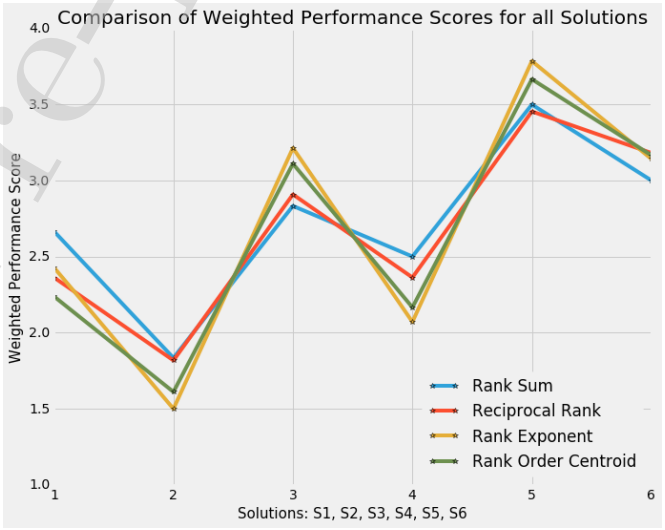


FIGURE 12. Comparative Evaluation of WPS of all the six solutions

It is clear from FIGURE 12 that the ideal recommended solution to reach the desired maturity of 60% ensuring the criticality of criteria and MIL of the controls are accounted for is solution *S5*. An interesting observation is the significant fluctuation across various rank-weights methods. It is clearly not an equal displacement across the solution's WPS values. Since the problem demonstrated has a very small amount of controls, all the methods converged at the same solution. However, it may be quite possible that if the number of controls is more realistic (at CSF level: 100+; C2M2 level: 300+), the converged solution may be different across the four depicted rank-weight methods.

Sri Nikhil Gupta Gourisetti
Sri Nikhil Gupta Gourisetti is a Grid – cybersecurity research engineer at Pacific Northwest National Laboratory (PNNL). During his research engagement at PNNL, he worked on several smart grid cyber-physical security projects addressing the security and grid systems interaction challenges and needs of critical facilities and infrastructure. He has been actively involved in research projects on security engineering solutions and responses for critical infrastructure. He is the Co-PI for DOE Cybersecurity projects such Keyless Infrastructure Security Solution – a blockchain-based system for critical infrastructure; Mitigation of External-exposure for Energy Delivery Systems; Cybersecurity Framework. He is one of the lead authors for DOE-PNNL led buildings cybersecurity framework and the lead developer for vulnerability assessment tools. He is the Co-PI for the development of on-going cyber-physical security non-intrusive applications. He is a technical lead for an incentive-based hardware-in-the-loop project where the grid simulation software and hardware systems interact in real-time. Under a DARPA project, he also led a team to develop red team – blue team cyber-physical attack scenarios for grid systems. He is pursuing his Ph.D in Engineering Sciences and Systems (Cybersecurity Framework and Algorithms for Prioritized Vulnerability Mitigation: An Adoption to Blockchain Systems) from University of Arkansas – Little Rock.

Dr. Michael Mylrea
Dr. Michael Mylrea is a Senior Advisor for Cyber Security and Blockchain Lead (PI) at Pacific Northwest National Laboratory. Dr. Michael has over 18 years of experience working on cyber security, energy and national security issues. This experience includes leadership positions in industry and government, including, but not limited to: U.S. Departments of Energy and Defense, Cyber Innovation Development (CSO & Co-Founder), Deloitte, U.S. Cyber Consequences Unit, Lakeside Oil, Harvard Berkman Center and Good Harbor Consulting. At PNNL, Michael leads several cyber security R&D and blockchain projects, including one of the first cybersecurity blockchain projects sponsored by the U.S. Department of Energy in partnership with Gaurdtime, Siemens, DoD and various energy companies. Michael is member of Washington State's IoT Council, an NSF Executive CyberCorps Fellow and a blockchain cyber security advisor to Rocky Mountain Institute. Michael recently completed his doctorate at George Washington University focused on cybersecurity resilience and organizational transformation

Dr. Hirak Patangia
Hirak Patangia is a professor in Systems Engineering, Engineering Technology, and Engineering Sciences and Systems at University of Arkansas – Little Rock. Throughout his career, Dr. Patangia have been leading various research projects in the areas of power electronics, electrical engineering, and signal processing under the grants funded by National Science Foundation. Dr. Patangia's work with electrical circuits and systems not only has theoretical applications, he has worked to use his research to develop practical concepts that are in demand in the private sector. For instance, his development work on Amplitude Division Multiplexing was assessed by the Arkansas State Highway Department for use in rural areas of the state. By using light-emitting diodes (LEDs) and solar panels, Dr. Patangia increased the effectiveness of reflectors used in construction zones to warn drivers of dangerous lane changes. After earning his bachelor's degree from the Indian Institute of Technology and his master's degree from the University of New Brunswick, Dr. Patangia completed work on his doctoral degree in electrical engineering from McGill University. Dr. Patangia served as program director for electronics engineering technology, as associate dean in the College of Science and Engineering Technology, and department chair in the College of Information Science and Systems Engineering. Currently, Dr. Patangia is the chair of Electrical and Computer Engineering Doctoral major under Engineering Sciences and Systems Department.

Sri Nikhil Gupta Gourisetti

Michael Mylrea

Hirak Patangia

Highlights

- Demonstrates a cybersecurity vulnerability mitigation framework (CyFEr) developed based on mathematical and logical constructs.
- The efficacy of the designed framework/tool/methodology is demonstrated by applying the framework on NIST Cybersecurity Framework (CSF).
- A real-world attack-based scenario is developed to validate and demonstrate the effectiveness of the integration of CyFEr with NIST CSF.
- A detailed applicability and comparative analysis are performed to evaluate the application of the presented framework. This is achieved by comparing the presented work against existing maturity models and quantitative vulnerability mitigation models.
- Multiple illustrations are used to demonstrate certain complex parts of the presented research.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: