IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 11, 2021, accepted April 14, 2021, date of publication April 19, 2021, date of current version April 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074021

A Bio-Inspired Reaction Against Cyberattacks: AIS-Powered Optimal Countermeasures Selection

PANTALEONE NESPOLI[1], **FÉLIX GÓMEZ MÁRMOL**[1], **AND JORGE MAESTRE VIDAL**[2], **(Member, IEEE)**
[1]Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain
[2]Digital Labs, Indra, 28108 Madrid, Spain

Corresponding author: Pantaleone Nespoli (pantaleone.nespoli@um.es)

This work was supported in part by the University of Murcia through the Formación del Profedorado Universitario (FPU) Predoctoral Contract, in part by the Ministerio de Economía y Empresa (MINECO), Spain, through the Ramón y Cajal Research Contract, under Grant RYC-2015-18210, in part by the European Social Fund, in part by the Spanish Ministry of Defence through the Cibermaniobras adaptativas y personalizables de simulación hiperrealista de APTs y entrenamiento en ciberdefensa usando gamificación (COBRA) Project under Grant 10032/20/0035/00, in part by the European Commission through Zero-touch Security and Trust for Ubiquitous Computing and Connectivity in 5G Networks (5GZORRO) under Grant 871533, and in part by the Practical Autonomous Cyberhealth for Resilient SMEs and Microenterprises (PALANTIR) Projects under Grant 883335.

ABSTRACT Nowadays, Information and Communication Technology (ICT) infrastructures play a crucial role for human beings, providing essential services at astonishing speed. Nevertheless, such a centrality of those infrastructures attracts the interest of ill-motivated actors that target such infrastructures with cyberattacks that are every day more sophisticated and more disruptive. In this alarming context, selecting the optimal set of countermeasures represents a primary need to react against the appearance of potentially dangerous threats effectively. With the motivation to contribute to developing faster and more effective response capabilities against them, the paper at hand introduces a novel cybersecurity reaction methodology based on Artificial Immune Systems (AIS), for which the evolutionary computing paradigm has been adopted. By leveraging the outstanding properties of these bio-inspired techniques, the selected countermeasures to defeat cyberthreats through cloning and mutation phases in an effort to improve the quality of the solution from a quantitative perspective, being able to adjust the risk to which the assets of the protected system are exposed. Exhaustive experiments demonstrate the feasibility of the proposed approach, reducing the risk in a more than acceptable time lapse.

INDEX TERMS Countermeasure selection, cyberattack countermeasures, intrusion reaction systems, artificial immune systems, bio-inspired reaction.

I. INTRODUCTION

The modern era brought a technological revolution never seen before. Human beings rely every day more on the services offered by powerful and responsive machines that significantly increase the quality of their lives [1]. In this direction, ICT infrastructures play an essential role in developing this modern society. That is, they permit to connect people among them at astonishing speed, offering also vital services as an ultimate goal. Additionally, the digital revolution is continuously breaking down barriers by proposing novel paradigms (i.e., Internet of Things (IoT) [2]) and disruptive technologies (i.e., Blockchain [3]) that are able to potentially change our lives.

Nevertheless, the significant advance witnessed in recent years also conveys some negative consequences. In fact, those crucial infrastructures are constantly targeted by attacks perpetrated by malicious entities, aiming at disrupting the availability of the provided services and threatening their confidentiality and integrity [4], [5]. Such ill-motivated entities are of a wide variety, from powerful institutions to skilled individuals, due also to the easiness of finding attack source code on the Internet, among other factors. Moreover, nations are getting more and more affected by this battle, with the conception of cyber warfare becoming a central matter within defense agencies while cyberspace consolidates as a fifth battle domain [6], [7].

Indeed, the security experts are registering a consistent increase in the number of cyberattacks that remarks the importance of posing cybersecurity as a fundamental pillar in the defensive strategies [8]. In such a frightening scenario, very few will oppose that selecting the optimal set of countermeasures to react against those threats is of primary importance. Remarkably, such a selection needs to balance the

The associate editor coordinating the review of this manuscript and approving it for publication was Vijay Mago.

inherent trade-off between the effectiveness of the reaction, aiming at eradicating the attack from the protected system and thus restoring a safe state and its potential negative impact (or side effect) on the targeted assets [9].

Until now, several proposals have been presented to solve the challenges posed by the reaction ecosystem. Concretely, a plethora of literature works proposes cost-benefit quantitative approaches in choosing the optimal set of countermeasures [10]. In some circumstances, those works leverage the useful capabilities of Artificial Intelligence (AI) methodologies and bio-inspired approaches to determine the optimal reaction in a semi-automatic or fully-automatic fashion [11].

Among those, AIS have been proved to comprise a promising paradigm in different areas of the cybersecurity ecosystem [12]. Specifically, such a bio-inspired technique aims to emulate the biological immune system's behavior in various applications of computer science. As the immune system is able to recognize and fight against foreign and potentially harmful entities to protect the human body, the AIS try to shield the monitored assets from possibly dangerous anomalies [13]. In this direction, AIS have been successfully applied to solve security-related challenges in the field of anomaly detection, intrusion detection, scan, and flood detection, among others [14]. Surprisingly, AIS capabilities have not been thoroughly investigated when it comes to selecting the optimal reaction to fire against appearing threats. Despite this fact, and as pointed out in [15], their properties applied to counteracting malicious cyber entities fit the needs of related reaction frameworks, which may weaponize biological principles like clonality, self-regulation, immune memory, specificity, and so forth.

In order to contribute to mitigating this gap, an AIS-powered reaction methodology is proposed, aiming at selecting and enforcing the optimal set of atomic countermeasures on the assets of the protected system exposed to risk. By leveraging the standard countermeasures representation presented in [16], the reaction phase entities are first translated to the artificial immunological ecosystem. Then, an index to evaluate the convenience of enforcing a particular atomic countermeasure on an asset is proposed, namely, the countermeasure benefit. Notably, such an index intends to capture the quantitative characteristics of a countermeasure, i.e., its effectiveness, impact, and cost. To this extent, since more than a countermeasure may be enforced on the same asset, a careful analysis is carried out to study from a quantitative perspective the combined effect of multiple countermeasures to be applied on such a same asset. Subsequently, the AIS-powered reaction is presented, detailing the steps needed to successfully respond and distinguish its properties for static and dynamic reactions. In particular, the antibodies are modeled as sets of atomic countermeasures that continuously pass through cloning and mutation phases in order to compute the optimal individual to adapt and fight against artificial antigens (namely, the threats detected within the protected system). To demonstrate the capability of the proposed methodology, a broad experimental phase is provided, resulting in minimizing the risk to which the assets are exposed in a more than acceptable time window.

For easy reference, the main contributions of the paper at hand can be summarized as follows:

- The proposal of a novel AIS-powered methodology to select the optimal set of countermeasures to react against potential cyberthreats. The adaptation of such a bio-inspired technique to the reaction field permits the acquisition of crucial AIS characteristics, such as *uniqueness*, *distributed reaction* and *self-regulation*, *diversification*, *self-protection*, and *memorization*. In particular, the presented methodology avoids minimizing risk blindly. That is, the calculated countermeasures are enforced taking into account the acceptable risk for each asset of the system.
- The AIS-powered reaction leverages a standard countermeasures representation, which we believe would be beneficial to foster the reaction knowledge sharing among different security teams.
- The proposition of a metric to quantitatively estimate the benefit of applying multiple countermeasures on the same asset, i.e., the *countermeasure benefit*.

The remainder of this paper is organized as follows. Section II presents some preliminary concepts on the AIS, reviews and discusses recent proposals related to countermeasures selection against threats, with particular focus on the proposed standard representation of a countermeasure. In Section III, the methodology used to construct the AIS-powered reaction is described, explaining the translation from cyberspace to the immunological sphere. Then, in Section IV, the AIS-powered reactions are analyzed, detailing the algorithms needed to achieve their purposes. Next, Section V presents the results of the conducted experiments, highlighting and discussing the advantages and drawbacks of the methodology. Finally, Section VI concludes the paper, proposing some interesting future lines to further contribute to the reaction ecosystem.

II. BACKGROUND

Although both academy and industry are focusing their effort around the cybersecurity context, some issues regarding the reaction ecosystems are still unsolved [10]. As a matter of example, potential tools to counteract ongoing threats are the Intrusion Reaction Systems (IRSs), which are IDSs (Intrusion Detection Systems) capable of reacting against suspicious activities in real or near real-time [17]. Additionally, recent efforts have been put to provide commercial SIEM (Security Information and Event Management) systems with automated response capabilities [5]. Those proposals indeed represent a significant advance in the endless arms race among malevolent entities and defensive teams, but one could say that there is still a long way to go.

Next, we analyze the most recent and relevant works regarding the countermeasures selection framework, highlighting their main properties and drawbacks. Then, we present some preliminary concepts to help readers fully understand the features of the AIS field, which motivate the presented methodology. Last but not least, the countermeasure standard proposal presented in [16] is summarized since it represents an essential basis on which this work is built.

A. RELATED WORKS

Within the cybersecurity field, the reaction phase against a cyberattack has been significantly less explored than the detection one, mainly due to the open challenges that still affect the response ecosystem, as well as a greater complexity at empirically validating the research outcomes [18], [19]. In this direction, the work in [10] analyzed the major reaction proposals from 2012 to 2017, highlighting their principal advantages and potential deficiencies.

Besides, in [20], the integration of a Stateful Return on Response Investment (StRORI) metric within Hypergraph models was proposed to effectively evaluate the potential application of countermeasures based on economic and threat assessment parameters. By solving the challenges affecting classic RORI-based models (i.e., do not consider the already deployed countermeasures), such metric is able to rank countermeasures also analyzing the previously implemented security measures. Then, the efficacy of the presented model is demonstrated by developing a prototype in a real use case scenario, using the attack graph modeling. Notably, the Common Vulnerabilities and Exposures (CVE)[1] and Common Attack Pattern Enumeration and Classification (CAPEC)[2] are leveraged as standard knowledge bases of vulnerabilities and attacks, respectively, to support their claims.

Additionally, a framework to respond to multi-path attacks is presented in [21]. Specifically, the authors formulated the problem of reacting to those attacks as an optimization problem, which appears to be *NP-hard*. To resolve such a problem, they proposed an ad-hoc greedy algorithm to select the most appropriate countermeasures in a cost-sensitive way. Authors leveraged the PART (Probabilistic Attack-Response Tree) models to represent potential attacker movements and evaluate three metrics: security benefit, deployment cost, and negative impact. Moreover, the feasibility of the proposed approach is later proved within a common virtual network presenting known CVE vulnerabilities.

Also, authors in [22] proposed an approach to implement a model-free IRS based on Deep Reinforcement Learning (DRL). To deal with the size of modern network infrastructures and their non-stationary characteristics, the methodology utilized DRL to find near-optimal responses in an acceptable time. More in detail, the system under protection is modeled as a set of components that possess state variables (i.e., active, updated, new version available, corrupted, vulnerable). The DRL learns from a simulated version of the real system and then is tested on it in a successive phase with a reward function based on execution time and cost of the actions executed. Experiments are added to compare the proposed DRL with Q-learning, focusing on non-stationary systems, demonstrating its feasibility.

Furthermore, a methodology to generate fine-grained response policies is presented in [23]. Starting from 4 fundamental questions about the reaction phase (i.e., which countermeasures should be selected, where should they be deployed, in what order multiple countermeasures are deployed, and how long do the countermeasures last), the authors proposed a decision-making framework for IRS that optimizes the responses based on four metrics (i.e., attack damage, deployment cost, negative impact, and security benefit). To solve such an optimization problem, a Genetic Algorithm with Three-dimensional Encoding (GATE) is considered, where a set of meta-policies represents each individual.

Moreover, the authors in [24] presented an innovative methodology for picking countermeasures based on AI techniques and the production of security metrics. The main goal is to select accurate responses to cyberattacks in near real-time, leveraging the data stemming from external security data sources and SIEM systems. To further define relationships among the various entities, an ontological strategy is introduced, joint with the logical inference used to extract knowledge. Then, the most probable attack path is predicted using an attack tree to deploy the optimal remediations.

Likewise, authors in [25] developed a procedure to achieve minimum cost defense in the context of Cyber-Physical Systems (CPS). In particular, such a procedure chooses optimal defense nodes using the Atom Attack Defense Trees (A2DT), which is a variant of the more conventional Attack-Defense Tree (ADT) model. Then, the authors used an ad-hoc methodology to solve the path calculation over the A2DT and demonstrated its applicability through 2 use cases (i.e., Automated Teller Machine (ATM) and Supervisory Control And Data Acquisition (SCADA) systems). Similarly, in [26], authors leveraged the ADT based on Directed Acyclic Graphs (DAGs) and then extracted from an ADT its defense semantics describing how the two actors (i.e., attacker and defender) may interact. Later, the ADT is translated into an Integer Linear Programming (ILP) problem that considers the various constraints (e.g., economic, actions of the actors, etc.). Notably, the authors developed an open-source tool to automate the described methodology.

The described works represent an important step within the reaction strategies ecosystem. Nevertheless, none of them leverages the significant advantages given by the adoption of a standard countermeasure representation. Moreover, very few of them feature the adaptability of the selected countermeasures against potential threats. It is clear that a methodology capable of adapting its response based on the asset over time would be extremely valuable when it comes to selecting the optimal set of countermeasures to counteract threats within the network, considering the resource availability at any time. Then, another important lack is highlighted in the literature, that is, the reaction frameworks

[1] https://cve.mitre.org/

[2] https://capec.mitre.org/

are applied to specific scenarios leveraging a comprehensive knowledge of the protected system. One could argue that, in order to be generic and thus applicable to several contexts, the countermeasure strategy should possess as little prior knowledge of the threats as possible. To this extent, the AIS-methodology is able to provide countermeasures by leveraging the non-determinism of the solution. In fact, the execution of the AIS-methodology may generate more than one applicable countermeasure so that human decision-makers can strategically select the best actions to undertake from a set of potentially effective solutions.

B. ARTIFICIAL IMMUNE SYSTEMS

Living beings have developed multiple immune mechanisms in an effort to battle against dangerous antigens harming their health [27]. Those immune mechanisms form part of the biological immune system, which features high distribution and parallelism within the organisms, with several cells, proteins, and organs participating [28].

Within the digital world, AIS represent the equivalent of the biological immune system to solve non-biological but computational problems. In fact, the AIS aim to emulate the mechanisms and behaviors observed by the immunologists in order to be applied to a particular problem [15]. After a quite complex modeling phase, AIS have been successfully applied to various fields of digital knowledge, including optimization theory [29], data analysis [30], image recognition [31], and computer security [32]. More specifically, AIS-based proposals have arisen to solve cybersecurity challenges for anomaly detection, intrusion detection, malicious process detection, scan and flood detection, and fraud detection, among others [33]–[36] others. Those proposals rely on the capabilities of four main bio-inspired techniques, namely, negative selection, clonal section, artificial immune networks, and Danger Theory [28], [37], [38].

Specifically, the clonal selection theory indicates that defined antibodies, in order to effectively counteract the malicious antigens, iteratively pass through various steps, namely: cloning, hypermutation, and selection phases. Throughout each iteration, the antibodies are further improved, getting closer to the optimal solution. In this direction, the CLONALG algorithm was proposed in [39], aiming at solving optimization and pattern recognition problems. Later on, this algorithm has been used to answer cybersecurity-related issues [40]. Also in this case, the modeling phase requires a notable effort since the main components of the problem that has to be solved need to be translated and encoded into immuno-related elements to apply such a bio-inspired technique.

As previously mentioned, several academic works proposed the application of AIS to solve open challenges within the cybersecurity field. In particular, those works leveraged the capabilities of such a bio-inspired methodology to detect a vast number of unknown patterns (*nonself* objects) from normal ones (*self* objects), often using limited resources [41]. Nevertheless, recent studies reveal that a remarkable research effort has been put on detecting the threats and that real-time mitigation or response has not been considered that much, though [15]. In this direction, it is worth remarking that the existing tools to counteract cyberattacks (i.e., the abovementioned IRS and new-generation SIEM) do not integrate AIS solutions within their capabilities.

C. COUNTERMEASURES STANDARD REPRESENTATION

One of the most prominent challenges within the reaction ecosystem is the lack of a standard countermeasures representation, whose direct consequence is the absence of a commonly shared and widely adopted knowledge base of remediations. To solve this issue, in our previous work [16], we contributed to this field by proposing a representation that details with fine granularity the fields needed to model the remediation object accurately. We believe that such a proposal is beneficial to foster the reaction knowledge sharing among different security teams, thus building more robust response plans as an ultimate goal.

$$cm_{ID} = (\underbrace{ID, Eff, Imp, Cost, \Omega_c, \Omega_p, \Omega_v, \Omega_a, \phi}_{mandatory}, \underbrace{P, \delta}_{optional}) \tag{1}$$

The proposed definition is shown in Equation 1, while its components are described in the following:

- $ID \in \mathbb{N}$ is the unique identifier of the countermeasure.
- $Eff \in \{L, M, H, X\}$ represents the residual effectiveness of the countermeasure, referring to its capability to neutralize the threat.
- $Imp \in \{L, M, H, X\}$ represents the residual impact of the countermeasure since it can negatively impact the corresponding asset(s) within the protected infrastructure.
- $Cost \in \mathbb{R}^+$ represents the residual cost of the countermeasure, which contains deployment, maintenance, and indirect costs.
- $\Omega_c = \{c_i \in C\}$ is the link to common configuration knowledge base C to which the designed asset refers.
- $\Omega_p = \{p_i \in P\}$ is the link to common platform knowledge base P to which the designed asset refers.
- $\Omega_v = \{v_i \in V\}$ is the link to common vulnerability knowledge base V to which the exploited vulnerability refers.
- $\Omega_a = \{a_i \in A\}$ is the link to common attack knowledge base A which the countermeasure counteracts.
- ϕ describes the enforcement of the countermeasure in a machine-readable format.
- $P = \{p_1, p_2, \ldots, p_n\}, n \geq 0$ describes the parameter(s) a certain countermeasure may need in order to be implemented.
- δ includes additional information about the countermeasure, particularly:
  - a textual description, in human-understandable language.

- a field indicating whether the deployment of the countermeasure demands software or hardware changes.
- a flag specifying if the countermeasure is static or dynamic.
- a field expressing whether the countermeasure is of short-term or long-term duration within the reaction strategy.
- examples of the enforcement (e.g., in pseudocode).
- if applied for cyber defense, linkage to military tactical, strategic, or operational decisions, which may provide structures describing, among others, related Courses of Action (CoA), command hierarchy, or mission-centric expected impacts.

In particular, the discrete values proposed for the residual effectiveness, impact and, cost stay for *Low (L), Medium (M), High (H)*, and *Not Defined (X)*, respectively. We indicate those values as residual since they represent intrinsic values of the countermeasure that persist during the time. Then, when a certain countermeasure has been selected to be enforced on a specific asset, it assumes real effectiveness, impact, and cost values, which also rely on other parameters which are connected to the countermeasure object itself, such as the dependencies among the corrupted services, the time needed to be implemented, or the perception of military-strategic planes, to cite some examples.

In addition to the presented fields, another parameter is introduced to further corroborate the reaction steps. That is, $\mathcal{M} \in [0, 1]$ defines the maturity of a previously-hardened countermeasure enforced within the system. Specifically, M takes into account how effective the enforcement of a security measure was on a certain asset of the system during a specific period. Obviously, if such enforcement was effective (e.g., by blocking an attack targeting an asset), the countermeasure tends to be considered as more *mature* and thus fired again against similar threats against similar assets. The maturity M of a countermeasure is calculated as the number of times the countermeasure has been effective N_s divided by the total number of implementations $N_s + N_f$, where N_f symbolizes the number of times the countermeasure failed to counteract the threat, as shown in Equation 2.

$$\mathcal{M}(cm_i) = \frac{N_s}{N_s + N_f} \tag{2}$$

III. METHODOLOGY

In the light of the above, an adaptation of the AIS algorithm is proposed, aiming to cherry-pick the optimal set of countermeasures to fire against a cyberthreat occurring within the protected system. It has to be remarked that the AIS-powered selection of countermeasures covers the entire spectrum of the reaction ecosystem. Specifically, it can be fired at a preventive phase due to its ability to prevent potential threats from happening, or at a reactive stage since it also is capable of eradicating an ongoing attack and remedying its adverse effects. In the context of this research, the authors distinguished between i) static countermeasure, referring to its preventive capabilities, and ii) dynamic countermeasure, indicating its reactive ones [16]. Thus, valuable features of the AIS methodology are imported to the reaction context, and listed in the following:

- **Uniqueness**: like in the immune system, each reaction here is unique against a specific threat.
- **Distributed reaction** and **self-regulation**: no central coordination and control are required during the reactive immuno-operation.
- **Diversification**: clonal selection and hypermutation constantly compute and present better responses to shield the system assets under attack.
- **Self-protection**: the reactive immuno-reaction protects nothing else but the designed assets, generating a tailored response while minimizing the negative impact.
- **Memorization**: reaction information is saved to optimize responses in the future.

By adopting the aforementioned features, one could easily say that the enforced reaction exhibits the desired properties in an effort to be optimal.

A. RATIONALE

Before analyzing the proposed AIS methodology to select the countermeasures, some fundamental concepts must be explained to understand further the cybersecurity context in which such immuno-algorithm is applied. Indeed, an appropriate modeling phase is fundamental to propose an AIS solution that can solve the problem correctly [15].

First, the system under protection can be modeled as a collection of several *assets*. Those assets present distinct software and hardware configurations and are deployed to execute various tasks. For instance, a web server that is in charge of serving requests coming from the Internet, a database that memorizes data of interest, or a firewall that filters the incoming connections, to name a few. It is worth noticing that the number of assets composing the ICT infrastructures nowadays is huge due to the central role of those systems in humans' everyday life.

$$A = \{A_1, A_2, \ldots, A_{N_A}\} \tag{3}$$

Specifically, Equation 3 clarifies the set of assets A of the entire system of cardinality N_A, where A_i represents a generic asset. Each of those assets possesses a value indicating its criticality $CR(A) \in [0, 1]$. Such a value indicates the importance of the asset for the network from a strategic or economic perspective, where 0 denotes a low critical asset, while 1 stands for a highly critical one. For example, the database storing the personal data of the employees can be considered of high value for an enterprise, thus the corresponding criticality value is expected to be high (e.g., $CR(database) = 0.9$).

Obviously, the assets of the systems are prone to expose security vulnerabilities. In this sense, the number and frequency of newly-discovered vulnerabilities are constantly increasing, posing a complex challenge to the CSIRT (Computer Security Incident Response Teams). Such growth is mainly due to the sophistication of modern assets but also to the high demand for new market-ready functionalities [1]. Vulnerabilities scanners normally report the vulnerabilities by means of vulnerabilities reports.[3] Thus, each asset of the network is connected to a *vulnerabilities set*, assuming that an asset can possess one or more vulnerabilities simultaneously.

$$V(A_x) = \{V_{A_x1}, V_{A_x2}, \ldots, V_{A_xl}\} \tag{4}$$

$$V = \{V_1, V_2, \ldots, V_{N_V}\} = \left(\bigcup_{x=1}^{N_A} V(A_x)\right) \tag{5}$$

Explicitly, each asset A_x of the system is associated with its vulnerabilities set $V(A_x)$ characterized by a certain number of vulnerabilities V_{A_xi}, as shown in Equation 4. To this extent, the union of the vulnerabilities set of all the assets is represented as V (see Equation 5), where V_i means a generic vulnerability within the entire system. Furthermore, the vulnerabilities of the assets can be potentially exploited by cyberattacks performed by ill-motivated entities, aiming at violating or interrupting their confidentiality, integrity, and availability (CIA). Such attacks are growing in complexity and power: multi-steps and zero-day attacks represent clear examples in this sense [21]. In this context, we denote ongoing attacks happening within the network with:

$$T = \{T_1, T_2, \ldots, T_J\} \tag{6}$$

Equation 6 refers to the attack set T composed by all the ongoing individual attacks T_i. Those attacks are unknown to the system before potential security incidents manifest, and they are detected by the security devices (e.g., IDSs, firewalls, etc.) deployed within the network. Those security devices, together with the vulnerability scanners, represent the *detectors* of the potential anomalies.

Both vulnerabilities V and attacks T that may appear within the network are considered *threats* against the assets. In particular, we denote such threats with τ (as reported in Equation 7). Referring to the AIS methodology, they represent the *antigens* against which the selection of countermeasures generates the response using *antibodies*.

$$\tau = T \cup V = T \cup \left(\bigcup_{x=1}^{N_A} V(A_x)\right) \tag{7}$$

To counteract the threats jeopardizing the assets of the system under protection (i.e., vulnerabilities or attacks), several countermeasures can be enforced during the reaction phase. That is, we pinpoint those remediation objects in a *countermeasure set CM*, representing the total number of countermeasures that can be enforced on the entire assets set A:

$$CM = \{cm_1, cm_2, \ldots, cm_N\} \tag{8}$$

Concretely, Equation 8 illustrates the countermeasure set CM constituted of N countermeasures cm_i. It has to be stated that the countermeasures belonging to the aforementioned set have to be considered as atomic objects, meaning that each one of them represents a fine-granularity reaction step. Besides, the countermeasures set features high dynamism within the reaction strategy since the included atomic countermeasures can be deleted (e.g., an individual asset or threat does not exist anymore in the system) or added (e.g., the appearance of new assets or threats) over time. Additionally, the countermeasures are stored following the standard representation described in Section II-C. Those objects represent the *antibodies* that the AIS-powered countermeasure system will select to fight against the *antigens*. Possible examples of countermeasures are "update software" in the context of a static reaction or "stop service" for a dynamic reaction.

More specifically, for each asset of the network, a subset of the entire countermeasures set is defined. Those remediations are tailored for a certain asset (e.g., software and hardware configuration, etc.), considering the vulnerabilities or the attacks that can threaten it. Besides, we assume that one or more countermeasures can be enforced simultaneously on the asset to fight against possible threats. Therefore, we denote the countermeasures set $CM(A_x)$ containing the atomic countermeasures cm_{A_xi} that can be enforced on the asset A_x in Equation 9; so $CM(A_x)$ can be understood as an action plan that describes the sequence of cm_{A_xi} countermeasures against the situation that triggered the execution of the bio-inspired algorithm. Precisely, each countermeasure set $CM(A_x)$ of the asset A_x is contained within the countermeasure set CM.

$$\begin{aligned} CM(A_x) &= \{cm_{A_x1}, cm_{A_x2}, \ldots, cm_{A_xj}\} \\ CM(A_x) &\subseteq CM \end{aligned} \tag{9}$$

B. COUNTERMEASURE BENEFIT

The selection of the optimal set of countermeasures has to consider the inherent tradeoff between the effectiveness of the remediation and its negative impact and cost [42]. Such balance burdens on the security administrator that has to maintain an adequate level of protection with a limited budget. Bearing this in mind, in the context of this research, we propose an index to evaluate the convenience of enforcing a certain atomic countermeasure cm_i. That is, the *countermeasure benefit* ($B \in [0, 1]$) uses the fields *Effectiveness*, *Impact*, *Cost* $\in [0, 1]$ fields of the standard representation in Section II-C to measure such convenience, as shown in Equation 10:

$$B(cm_i) = \omega \times (Eff(cm_i))^{1-\left(\frac{Imp(cm_i)+Cost(cm_i)}{2}\right)} \tag{10}$$

Nevertheless, $B(cm_i)$ is shifted to reside in the interval $[1, 10]$ by adding ω in Equation 10 to preserve the compatibility with other security scoring systems. Consequently, if the enforcement of a certain atomic countermeasure does not worsen the security attributes of the assets, then $B > 1$. By default, the value $B = 1$ is reserved to the special countermeasure "no action" featuring the lowest B

[3] https://www.bmc.com/blogs/vulnerability-reports/

(i.e., $B(no\ action) = 1$). Note that the proposed B considers the composing parameters as equally important. Nonetheless, weights may be added to the equation to give more significance to a certain parameter. For instance, *Cost* can be more relevant in situations where the budget constitutes a considerable limitation. For a quick reference, Figure 1 illustrates the trend of the parameters composing benefit B in a 3D graph.

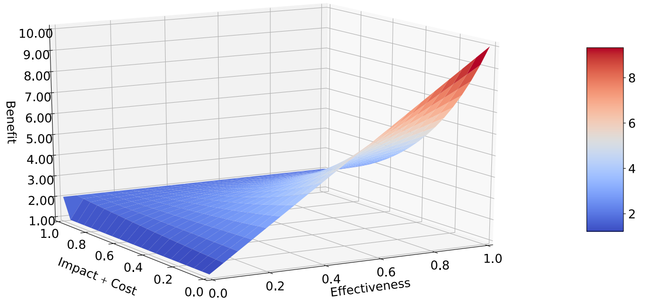


FIGURE 1. **Benefit of a countermeasure cm_i, $B(cm_i)$, based on its effectiveness, impact, and cost.**

As previously mentioned, one or more countermeasures can be simultaneously enforced on a single asset. To this extent, a crucial point is the study of how the atomic countermeasures combine their effect in an effort to determine which ones should be activated. Recent works propose to compute the joint effect of countermeasures by using the attack surface or the vulnerabilities coverage [43]. Nevertheless, such a methodology assumes previous knowledge of potential vulnerabilities present within, or attacks against, the protected system. In this work, we adopt a *defensive* perspective with no former knowledge of vulnerabilities or attacks till they are detected. The main reasoning behind this choice lies in the fact that the countermeasures selection system needs to be as generic as possible, so it would be possible to apply the proposed methodology to different scenarios.

Thus, given two candidate atomic countermeasures cm_i and cm_j that must be enforced on the same asset, evaluating the combination of the effects of those countermeasures is essential. On the one hand, if the enforcement of cm_j does not improve the effects arising from the enforcement of cm_i, then one infers that the countermeasures are not combinable. To this extent, the combined benefit B of the enforcement of cm_i and cm_j results to be the maximum of the benefit among those, as shown in Equation 11:

$$B(\{cm_i, cm_j\}) = max(B(cm_i), B(cm_j)) \quad \text{if } cm_i, cm_j \text{ are not combinable} \tag{11}$$

Such a situation represents the worst possible combination since we assume that the implementation of a security measure does not decrease the total benefit. More specifically, the enforcement of an atomic countermeasure generates a benefit $B > 1$ that does not interfere with the enforcement of another one. Then, the goodness of an antibody (i.e., a set of atomic countermeasures) is refined during the AIS-powered algorithm of reactions, as we will see in Section IV. Thus, the combined value defined in Equation 11 establishes the *lower bound* of the combined benefit of two countermeasures.

On the other hand, if the enforcement of cm_j definitely enhances the effects generated by the enforcement of cm_i, one says that the countermeasures are perfectly combinable. In this case, the combined benefit B of the enforcement of cm_i and cm_j can be defined as the sum of the single benefit of those countermeasures limited by the maximum possible value of this parameter (i.e., 10), as reported in Equation 12:

$$B(\{cm_i, cm_j\}) = min(B(cm_i) + B(cm_j), 10) \quad \text{if } cm_i, cm_j \text{ are perfectly combinable} \tag{12}$$

This condition expresses the best possible combination, so the combined value illustrated in Equation 12 determines the *upper bound* of the combined value of two countermeasures.

Consequently, the value of the combined benefit $B(\{cm_i, cm_j\})$, based on the combinability of cm_i and cm_j, can be defined as:

$$\underbrace{max(B(cm_i), B(cm_j))}_{\text{if } cm_i, cm_j \text{ are not combinable}} \leq B(\{cm_i, cm_j\}) \leq \underbrace{min(B(cm_i)+B(cm_j), 10)}_{\text{if } cm_i, cm_j \text{ are perfectly combinable}} \tag{13}$$

That is, Equation 13 states that the benefit $B(\{cm_i, cm_j\})$ of two countermeasures swings in a range depending on how much the effects of those reaction steps are combinable, being limited by the lower and upper bounds.

At this point, it is clear that the knowledge on how much the effect of two countermeasures can be combined is extremely valuable in an effort to calculate the combined benefit B. Nonetheless, one could easily argue that the analysis of all possible combinations of enforcement of atomic countermeasures, against all potential threats (during the static or dynamic reaction) is not viable. Moreover, the set of atomic countermeasures is highly dynamic, as remarked before. Additionally, as mentioned before, the proposed countermeasures selection methodology should be as generic as possible. To tackle this challenge, we propose a simple but effective approach to compute the combined benefit $B(\{cm_i, cm_j\})$ of two countermeasures cm_i and cm_j, defined as the midpoint of the interval illustrated in Equation 13:

$$B(\{cm_i, cm_j\}) = \frac{max(B(cm_i), B(cm_j)) + min(B(cm_i)+B(cm_j), 10)}{2} \tag{14}$$

Generalizing Equation 14 to N possible countermeasures in the countermeasures asset CM, the benefit of enforcing N atomic countermeasures on a specific asset is expressed in

Equation 15:

$$B(\{cm_1, cm_2, \ldots, cm_N\}) = \frac{max_{1 \leq i \leq N}(B(cm_i)) + min\left(\sum_{i=1}^{N} B(cm_i), 10\right)}{2} \tag{15}$$

C. RISK LEVEL EVALUATION

The selection of the optimal set of countermeasures, and its subsequent enforcement, aims at protecting the assets of the system by tuning the risk level as an ultimate goal. Hence, we define the *risk level* $RL(\tau_k, A_x, CM_j(A_x)) \in [0, 10]$ (where $RL = 0$ represents a situation of no risk, while $RL = 10$ expresses an extremely-high risk one) of a certain asset A_x, facing the threat τ_k, and protected with the set of atomic countermeasures $CM_j(A_x)$, as:

$$\begin{aligned} RL(\tau_k, A_x, CM_j(A_x)) &= \frac{P(\tau_k, A_x) \times I(\tau_k) \times CR(A_x)}{B(CM_j(A_x))} \\ s.t.\ 0 &\leq P(\tau_k, A_x) \leq 1 \\ 1 &< I(\tau_k) < 10 \\ 0 &< CR(A_x) < 1 \\ 1 &\leq B(CM_j(A_x)) \leq 10 \end{aligned} \tag{16}$$

In Equation 16, several parameters are considered to calculate the risk level correctly. First, $P(\tau_k, A_x)$ represents the probability of the occurrence of the threat τ_k over the asset A_x. Second, $I(\tau_k)$ expresses the negative impact of the threat τ_k on the asset A_x or the normal network operations. Then, the criticality of the asset $CR(A_x)$ is added since we assume that the more critical the asset is, the higher the risk level. In an effort to reduce the risk, a set of countermeasures can be potentially enforced, generating a benefit $B(CM(A_x))$.

In this direction, enforcing a set of countermeasures on a specific asset should avoid risk minimization blindly. In fact, the implementation of security measures should prevent both the underprotection or overprotection conditions over the assets of the system. For instance, excessive enforcement of countermeasures may result, on the one hand, in overprotecting the assets, minimizing the risk more than necessary (and possibly impacting their usability and consuming more resources). Thinking to scenarios in which the availability of the resources is acutely low and controlled (e.g., IoT scenario), employing more resources than the ones that are effectively needed is not recommended. On the contrary, an incorrect selection of countermeasures may culminate in underprotecting the assets, exposing them to concrete threats, which in critical scenarios may cause high risks for the entire system. To solve such a problem, the *acceptable risk level* $\widetilde{RL}(A_x) \in [0, 10]$ is assigned to each asset of the network A_x, and is defined as:

$$\widetilde{RL}(A_x) = \alpha \times (1 - CR(A_x)) \tag{17}$$

As seen in Equation 17, such an important parameter is directly derived from the criticality $CR(A_x)$ of the asset A_x by adding the scaling factor $\alpha = 10$. In particular, if the asset A_x is highly critical (e.g., $CR(A_x) \approx 1$), the corresponding acceptable risk level is extremely low (e.g., $\widetilde{RL}(A_x) \approx 0$). This means that, in case of the appearance of any threat jeopardizing A_x, the countermeasures selection strategy shall enforce efficient security measures to reduce the risk level $RL(A_x)$ to $\widetilde{RL}(A_x)$. On the contrary, if A_x possesses a low criticality value (e.g., $CR(A_x) \approx 0$), the acceptable risk level is quite high (e.g., $\widetilde{RL}(A_x) \approx 10$). In this case, if any threat appears, the selection methodology may decide not to enforce countermeasures at all or remove them from the selection.

Bearing this in mind, the optimal selection of countermeasures is in charge of minimizing the difference between the measured risk level RL and the acceptable risk level $\widetilde{RL}$ of each asset A_x of the system threatened by τ_k, as shown in Equation 18. From now on, function f is referred to as *fitness function*.

$$\min_{CM_j(A_X)} f = |RL(\tau_k, A_x, CM_j(A_x)) - \widetilde{RL}(A_x)| \tag{18}$$

Particularly, since the value of the fitness function belongs to the interval [0, 10], Equation 18 directly indicates that a lower value of fitness implies a better solution. It is worth remarking that, in the case of a static reaction, the threat τ_k is replaced by a vulnerability $V_{A_x k}$, while, during the dynamic reaction, τ_k is represented by an attack T_k. Moreover, when it comes to cherry-picking the correct countermeasures, it is possible to leverage the flag presented within the countermeasures standard representation in Section II-C that indicates whether the countermeasure features static or dynamic characteristics.

IV. AIS-POWERED REACTION

As previously mentioned, the AIS-powered selection of countermeasures embraces both static and dynamic reactions. Throughout this Section, particular focus is given to the methodologies to effectively react to malicious events threatening the assets of the system under protection. In particular, the algorithms presented next must be considered part of the paper's contribution since they represent a novel adaptation of the AIS to the reaction ecosystem. For a quick reference, Figure 2 illustrates the flow of events and the main components involved within the static and dynamic reactions that will be detailed next.

A. AIS STATIC REACTION

If any dangerous vulnerability is exposed by an asset within the network, the AIS static reaction is in charge of intervening in a preventive fashion. Recalling Equation 16, the risk level for the AIS static reaction can be defined as follows:

$$RL(V_{A_x k}, A_x, CM_j(A_x)) = \frac{P(V_{A_x k}, A_x) \times I(V_{A_x k}) \times CR(A_x)}{B(CM_j(A_x))} \tag{19}$$

Specifically, the vulnerability $V_{A_x k}$ represents the antigen against which the antibodies (i.e., set of atomic countermeasures $CM_j(A_x)$ of the asset A_x) must be selected. Besides,

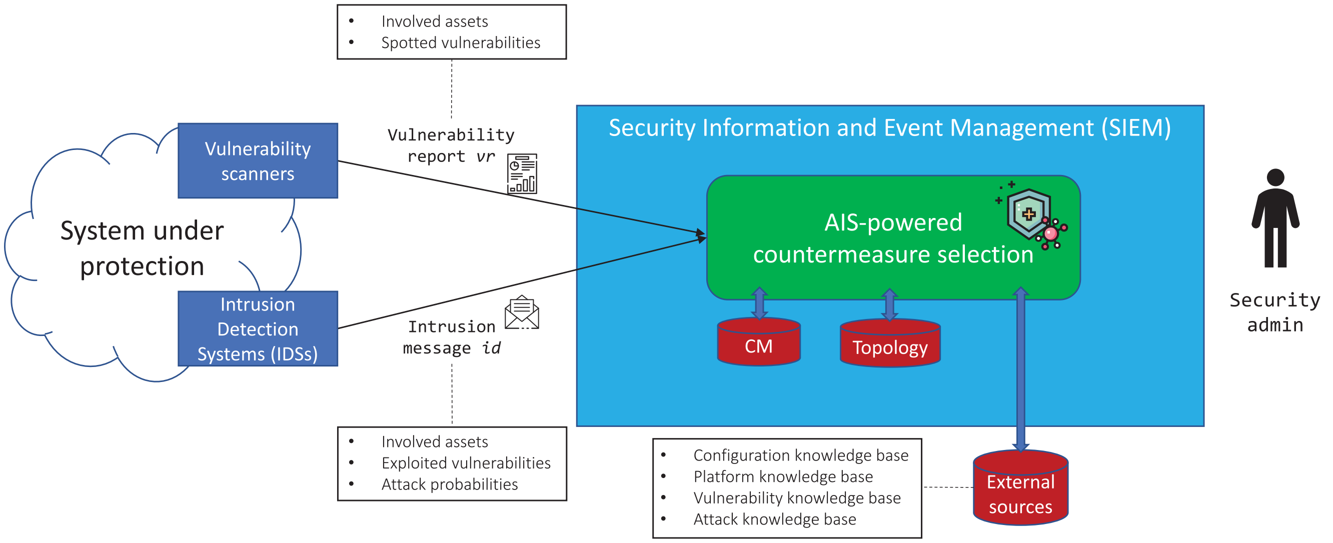


FIGURE 2. High-level architecture for AIS-powered reactions.

$P(V_{A_x k}, A_x)$ represents the probability that the vulnerability $V_{A_x k}$ of the asset A_x is exploited. Then, $I(V_{A_x k})$ measures the impact that the exploitation of the vulnerability $V_{A_x k}$ has on the asset and, consequently, on the system. Both $P(V_{A_x k}, A_x)$ and $I(V_{A_x k})$ can be directly derived from Common Vulnerability Scoring System (CVSS[4]) fields related to $V_{A_x k}$. Since an asset can expose one or more vulnerabilities simultaneously, a risk level calculation must be computed for each vulnerability. Thus, the main goal of the AIS static reaction is to calculate the optimal set of countermeasures CM_j that minimizes the fitness function f. More specifically, for each asset A_x of the system exposing vulnerabilities $V(A_x)$, such a function is calculated as the difference between (i) the sum of the measured risk levels RL generated by the vulnerabilities divided by the number of vulnerabilities of the asset $V(A_x)$ and (ii) the acceptable risk level $\widetilde{RL}$ of those assets, normalized by N_A, as shown in Equation 20:

$$\min_{CM_j} f = \frac{\sum_{x=1}^{N_A} \left| \frac{\sum_{V_i \in V(A_x)} RL(V_i, A_x, CM_j(A_x))}{|V(A_x)|} - \widetilde{RL}(A_x) \right|}{N_A} \tag{20}$$

The pseudocode of the AIS static reaction is reported in Algorithm 1. To this extent, we assume that a certain number of vulnerabilities scanners $S \in \mathbb{S}$ are deployed within the network. Those tools are in charge of scanning the assets of the network with a determined frequency (e.g., once per hour) to spot potential vulnerabilities. Concretely, lines 1-7 describe the task of the vulnerabilities scanners S in search of potential vulnerabilities V_i, which perform as antigens in the proposed static version of AIS methodology. Whenever a previously unknown vulnerability is found, a vulnerability report vr_i is created and sent to the SIEM for further analysis. In particular, this report contains vital information such as the asset that exposes the vulnerability, the ID, and the severity of the vulnerability, among others. A widely-used vulnerability scanner is OpenVAS,[5] an open-source and powerful vulnerability assessment tool capable of vulnerability scanning and management. It identifies the active services, open ports, and running applications across the machines.

Furthermore, in lines 8-14, the SIEM receives the vulnerability reports and inspects the information contained in them. Specifically, for each asset A_x that presents a vulnerability V_i, the correlated parameters $P(V_i, A_x)$, $I(V_i)$, and $CR(A_x)$ are extracted (line 10) in an effort to determine the risk level RL associated with the specific vulnerability in the absence of implemented countermeasures (line 12). Bear in mind that $B = 1$ in case of "no action", that is, the absence of implemented countermeasures, as initialized in line 11.

Then, lines 15-22 detail the AIS static methodology to select the optimal set of countermeasures. More in detail, an initial set of random solutions $\mathbb{CM}$ is generated (line 15). This set contains the antibodies used to fight against the antigens (i.e., vulnerabilities). Each one of the antibodies CM_j is, in turn, a set of atomic countermeasures to be implemented on each asset A_x that exposes a vulnerability V_i. In order to improve the initial generation of the solution, it is possible to leverage the *maturity* field $\mathcal{M}$ (Equation 2). That is, if any of the atomic countermeasures possesses a high maturity score against a particular vulnerability, it can have a higher probability of belonging to the initial set of solutions.

Once the initial generation has been performed, the affinity of the antibodies is calculated (line 17). The pseudocode of this function is reported in Algorithm 2. In particular, for each asset A_x presenting a vulnerability V_i and for each atomic countermeasure cm_k assigned to the solutions $CM_j \in \mathbb{CM}$, the benefit B of the countermeasures (Equation 10) and the

[4] https://www.first.org/cvss/

[5] https://www.openvas.org/

Algorithm 1 `AIS Static Reaction`($\mathbb{S}, CM$)

```
Require: $\mathbb{S} \neq$ null                                        ▷ Active vulnerability scanners
Require: $CM \neq$ null                                        ▷ Countermeasures knowledge
 1: for all $S \in \mathbb{S}$ do
 2:     Scan the network to spot vulnerabilities
 3:     for all $V_i$ detected & $V_i \notin V$ do                ▷ If any new vulnerability is detected
 4:         Create vulnerabilities report $vr_i$ and send it to the SIEM
 5:         $V \leftarrow V \cup \{V_i\}$
 6:     end for
 7: end for
 8: for all $V_i \in V$ do                                ▷ SIEM analyzes the vulnerabilities reports
 9:     for all assets $A_x$ exposing $V_i$ do
10:         Extract the associated parameters $P(V_i, A_x), I(V_i), CR(A_x)$
11:         $CM(A_x) = \{no\ action\}$
12:         Compute $RL(V_i, A_x)$                    ▷ Risk level calculation with no $cm$ enforced
13:     end for
14: end for
15: Generate initial random solutions $\mathbb{CM} \leftarrow \{CM_1, CM_2, \ldots\}$   ▷ Antibodies (sets of atomic cms)
16: while stop condition is not met do
17:     Determine affinity of Antibodies sets                             ▷ Evaluation of the risk
18:     Clone Antibodies with best affinity         ▷ Number of clones proportional to the affinity
19:     Mutate attributes of the clones                          ▷ Add or remove cms from CM
20:     Replace antibodies with lowest affinity                        ▷ Remove bad solutions
21:     $\overline{CM} \leftarrow$ Best Antibody
22: end while
23: for all $cm \in \overline{CM}$ do                           ▷ Atomic $cm$ in the best solution
24:     Update links to external security knowledge bases
25:     Update $\mathcal{M}$                                              ▷ Update Maturity
26:     $V \leftarrow \emptyset$
27: end for
```

RL (Equation 19) are calculated. To this extent, if more countermeasures have been selected as a solution to be enforced on the same asset A_x, the combined benefit $B(CM_j(A_x))$ must be evaluated as shown in Equation 15. Then, for each $CM_j \in \mathbb{CM}$, the corresponding fitness function $f(CM_j)$ is calculated as illustrated in Equation 20. Besides, the average affinity of the antibodies is computed.

Later, in line 18, the antibodies with the highest affinity (i.e., sets of atomic countermeasures which minimize the function f) are cloned, as shown in Algorithm 3. Concretely, the solutions $CM_j \in \mathbb{CM}$ are ordered by increasing fitness function f, and consequently, the K best CM_j are cloned, thus expanding the solutions space $\mathbb{CM}$. At this stage, a crucial point is the correct selection of K. On the one hand, a reduced number of clones may avoid creating a broad population (which leads to diversity in the solution). On the other hand, the generation of several clones may slacken the convergence of the algorithm toward acceptable solutions. In this direction, a potential improvement is the selection of K proportional to the fitness function f. For instance, if such a function has not been sufficiently minimized (because of the low affinity of the antibodies), the number of clones may be higher to further explore the solutions space to search for better solutions.

Afterward, the produced K clones $\{CM_{j_1}, \ldots, CM_{j_K}\}$ are mutated (line 19), as illustrated in Algorithm 4. More specifically, each set of atomic countermeasures (i.e., cloned antibody) undergoes through a mutation consisting of adding or removing an atomic countermeasure from the set with a certain probability P. To this extent, the proposed mutation adds or removes one atomic countermeasure from a clone CM generating CM' with the same probability $P = 0.5$. Then, the affinity of the mutated clones $f(CM')$ is computed (i.e., the fitness is evaluated). If such affinity is greater than the average affinity of the antibodies in the solution space $avg\_affinity(\mathbb{CM})$ (i.e., the atomic countermeasures of the clones are not effective against the vulnerabilities), then those clones are removed. Otherwise, they become part of the solutions set $\mathbb{CM}$, replacing the lowest affinity antibodies in $\mathbb{CM}$ at that instant.

Next, in line 20, the antibodies with the lowest affinity are removed from the solution space, as reported in Algorithm 5. In particular, the K sets of atomic countermeasures that exhibit the lowest value for the fitness function f are eliminated from the solutions set $\mathbb{CM}$, and, therefore, they are replaced with K random sets. Also in this case, as for the initial random generation

Algorithm 2 Determine_Affinity($V, A, \mathbb{CM}$)

```
 1: for all $V_i \in V$ do                                        ▷ For each detected vulnerability
 2:     for all assets $A_x \in A$ exposing $V_i$ do              ▷ For each involved asset
 3:         for all $CM_j \in \mathbb{CM}$ do                     ▷ For each antibody
 4:             for all $cm_k \in CM_j$ do                        ▷ For each atomic countermeasure
 5:                 Calculate $B(cm_k)$
 6:             end for
 7:             if $\exists\ CM_j(A_x) \subseteq CM_j$ s.t. $|CM_j(A_x)| > 1$ then   ▷ If two or more atomic cms enfornced on the same asset
 8:                 Calculate combined benefit $B(CM_j(A_x))$
 9:             end if
10:             Evaluate the risk $RL(V_i, A_x, CM_j(A_x))$       ▷ Calculate risk of each antibody
11:         end for
12:     end for
13: end for
14: for all $CM_j \in \mathbb{CM}$ do
15:     Evaluate $f(CM_j)$                                        ▷ Compute fitness of each antibody
16: end for
17: $avg\_affinity(\mathbb{CM}) \leftarrow \sum_{CM_j \in \mathbb{CM}} f(CM_j)/|\mathbb{CM}|$   ▷ Average affinity of the antibodies
```

Algorithm 3 Clone_Antibodies($\mathbb{CM}$)

```
1: for all $CM_j \in \mathbb{CM}$ do
2:     Order by $f(CM_j)$
3: end for
4: Clone $K$ best $CM_j$, $\{CM_{j_1}, \ldots, CM_{j_K}\}$      ▷ $K$ proportional to fitness function $f$
```

Algorithm 4 Mutate_Clones($\{CM_{j_1}, \ldots, CM_{j_K}\}$)

```
 1: for all $K$ clones $CM \in \{CM_{j_1}, \ldots, CM_{j_K}\}$ do
 2:     $P \leftarrow random(0, 1)$
 3:     if $P > 0.5$ then                                         ▷ With probability $P = 0.5$
 4:         Add random $cm$ to $CM$ to produce $CM'$
 5:     else                                                      ▷ With probability $\neg P = 0.5$
 6:         Remove random $cm$ to $CM$ to produce $CM'$
 7:     end if
 8:     Evaluate $f(CM')$                                         ▷ Determine the affinity of the mutated clone $CM'$
 9:     if $f(CM') > avg\_affinity(\mathbb{CM})$ then
10:         Discard mutated clone $CM'$
11:     else
12:         Replace lowest affinity antibody in $\mathbb{CM}$ with $CM'$
13:     end if
14: end for
```

Algorithm 5 Replace_Antibodies($\mathbb{CM}$)

```
1: Eliminate $K$ lowest affinity $CM$ from $\mathbb{CM}$
2: Add $K$ randomly generated $CM$ to $\mathbb{CM}$
```

(line 15), the maturity $\mathcal{M}$ of the countermeasures may be considered.

Finally, the best antibody is assigned to the final solution (line 21). The loop of lines 16-22 of Algorithm 1, containing the AIS static reaction methodology, continues till the stop condition is not met. Thus, it is clear that the choice of a correct stop condition is fundamental to execute an adequate number of iterations. In this direction, since the AIS static reaction is executed in a preventive fashion (i.e., the vulnerabilities are present within the system but are not currently under exploitation), it is coherent to assume that the algorithm can be run off-line and with a sufficient number of iterations. Besides, the stop condition can be further enriched by adding finer-granularity options, such as the acceptable risk level $\widetilde{RL}$, timing conditions, or a combination of those.

Thus, the selected antibody possesses the best fitness value, minimizing the difference between the measured and the acceptable risk levels. The atomic countermeasures included in the antibody may be presented to the human decision-maker (e.g., the system administrator) that is in charge of selecting its enforcement based on the most suitable strategy for the system. In the case that the proposed solution does not satisfy the criteria (e.g., low quality of the solution due to incorrect setting of the stop conditions), the decision-maker may decide to relaunch the algorithm assuming a more resource-consuming execution (in terms of time or memory, for instance), which may have a greater likelihood of finding better results due to the non-determinism of the algorithm.

Once the loop in lines 16-22 terminates its execution, and consequently, the best antibody has been found, the atomic countermeasures composing the solution undertake an updating process (lines 23-27). Following the principles of the standard representation in Section II-C, the links to the external common security knowledge bases are updated, as well as the maturity of the atomic objects. For instance, if the selected countermeasures have successfully covered a set of vulnerabilities, a connection with the external knowledge base of vulnerabilities is created (e.g., a link with the CVE). Then, the vulnerability set V is emptied.

B. AIS DYNAMIC REACTION

Whenever an ongoing attack actively targets an asset of the network, the AIS dynamic reaction is fired in an effort to eradicate it. Regarding Equation 16, the risk level in the case of dynamic reaction can be characterized as follows:

$$RL(T_k, A_x, CM_j(A_x)) = \frac{P(T_k, A_x) \times I(T_k) \times CR(A_x)}{B(CM_j(A_x))} \tag{21}$$

More in concrete, the attack T_k represents the antigen against which the antibodies (i.e., the set of atomic countermeasures $CM_j(A_x)$ of the asset A_X) must be selected. In addition, $P(T_k, A_x)$ expresses the probability that the attack T_k is currently affecting the asset A_x. Contrary to the AIS static reaction, such a probability can be directly acquired by the security devices (e.g., IDS, firewall, etc.) which reported the attack. Then $I(T_k)$ measures the impact of the ongoing attack on the targeted asset and, subsequently, the entire system. In this case, $I(T_k)$ can be determined from the CVSS field related to the vulnerability that the attack is exploiting. It has to be stated that we assume that a certain asset can be targeted by one specific attack at a time.

For the dynamic reaction, some additional concerns need to be considered. First, since a particular asset of the network has been compromised, the AIS selection of countermeasures has to protect this asset but also the connected ones (logically or physically). We refer to those assets connected with A_x, plus A_x itself, as $Y(A_x) = \{A_x, A_{x_1}, A_{x_2}, \ldots, A_{x_y}\}$. This choice lies in the fact that, under this situation, it is coherent to assume that the attacker may target other assets in an escalation to reach his/her ultimate goal [44]. That is, in the context of multi-step attack scenarios, CVSS scoring may be framed within the environmental threat conditions (e.g., collateral damage potential, target distribution, etc.). To achieve this goal and, consequently, counteract multi-step attacks, the AIS methodology can leverage high-quality attack models such as attack graphs [45] or service dependency graphs [46] to evaluate which other assets of the network present a higher risk and with which probability. Such a probability is also used to compute Equation 21 when it comes to protecting the connected assets. Besides, by looking at the standard representation in Section II-C, it is possible to leverage the *parameters* presented in the optional fields. Specifically, for certain countermeasures, some parameters can be defined in order to be executed on the targeted asset. With this field, the *parametric* countermeasure is enforced using only the amount of resources that are effectively needed to respond to the threat. Examples of combinations of reaction steps and parameters are, say, "reduce bandwidth by 50%" or "block IP address for 30 minutes", and so forth. Thus, a *parametric countermeasure* can be formally defined as:

$$cm_{A_x i, p_k} = \{cm_{A_x i} | p = p_k\} \tag{22}$$

Equation 22 describes that a certain atomic countermeasure $cm_{A_x i}$ belonging to the countermeasures set of the asset A_x is parametric, and its enforcement parameter is p_k.

Hence, the main objective of the AIS dynamic reaction is to compute the optimal set of countermeasures $CM_j(A_x)$ that minimizes the fitness function f. This function is computed as the difference between (i) the measured risk level RL generated by the ongoing attack T_k on the assets belonging to $Y(A_x)$ and (ii) the acceptable risk level $\widetilde{RL}$ of those assets, normalized by $|Y(A_x)|$, as depicted in Equation 23.

$$\min_{CM_j} f = \frac{\sum_{A_x \in Y(A_x)} \left| RL(T_k, A_x, CM_j(A_x)) - \widetilde{RL}(A_x) \right|}{|Y(A_x)|} \tag{23}$$

Algorithm 6 contains the pseudocode of the AIS dynamic reaction. In this direction, we assume that a finite number of IDSs $I \in \mathbb{I}$ (or related cyber sensing capabilities, services, functions, etc.) are deployed within the system. Those devices are in charge of detecting potential intrusions which can target a specific asset. In particular, lines 1-6 describe the job of the IDSs I hunting for ongoing attacks T, which act as antigens in the presented algorithm. Whenever a new attack is detected, an intrusion detection message is shaped and sent to the SIEM for further investigation. To this extent, the intrusion detection message encapsulates meaningful information about the attack, such as the targeted asset, the exploited vulnerability, the source, and so forth. In this direction, Intrusion Detection Message Exchange Format (IDMEF) [47] and Incident Object Description Exchange Format (IODEF) [48] can be seen as the *de facto* standards to exchange intrusion detection messages.

Moreover, in lines 8-14, the SIEM analyzes the received messages. Concretely, for each asset A_x included in $Y(A_x)$ (i.e., targeted asset A_x and connected ones), the corresponding

Algorithm 6 `AIS Dynamic Reaction`($\mathbb{I}$, CM)

```
Require: $\mathbb{I} \neq$ null                                          ▷ Active IDSs
Require: $CM \neq$ null                                                  ▷ Countermeasures knowledge
 1: for all $I \in \mathbb{I}$ do
 2:     Monitor the network to detect intrusion
 3:     for all $T_k$ detected & $T_k \notin T$ do                      ▷ If any new intrusion is detected
 4:         Send intrusion detection message $id_i$ to SIEM
 5:         $T \leftarrow T \cup \{T_k\}$
 6:     end for
 7: end for
 8: for all $T_k \in T$ do                                              ▷ SIEM analyzes the intrusion detection messages
 9:     for all asset $A_x \in Y(A_x)$ do
10:         Dynamically extract the associated parameters $P(T_k, A_x)$, $I(T_k)$, $CR(A_x)$
11:         $CM(A_x) = no\ action$
12:         Compute $RL(T_k, A_x)$                                      ▷ Risk level calculation with no cm enforced
13:     end for
14: end for
15: Generate initial random solutions $\mathbb{CM} \leftarrow \{CM_1, CM_2, \ldots\}$   ▷ Antibodies (sets of atomic cm)
16: while stop condition is not met do
17:     Determine affinity with Antibodies sets                         ▷ Dynamic evaluation of the risk for the assets
18:     Clone subset of Antibodies with best affinity                   ▷ Number of clones proportional to the affinity
19:     Mutate attributes of the clones                                 ▷ Add or remove cms, or modify parameters
20:     Replace low affinity antibodies                                 ▷ Remove bad solutions
21:     $\overline{CM} \leftarrow$ Best Antibody
22: end while
23: for all $cm \in \overline{CM}$ do                                   ▷ Atomic cms in the best solution
24:     Update links to external database
25:     Update $\mathcal{M}$                                            ▷ Update Maturity
26:     $T \leftarrow \emptyset$
27: end for
```

parameters $P(T_k, A_x)$, $I(T_k)$, and $CR(A_x)$ are extracted (line 11), and the associated risk level RL in the absence of countermeasures (i.e., $CM(A_x) = no\ action$) is calculated (line 12). During the dynamic reaction, it has to be remarked that timing is one of the most critical factors. Therefore, each step needs to be executed in a timely fashion. Next, lines 15-22 contain the AIS dynamic methodology to select the optimal set of countermeasures. Similarly to the AIS static reaction, an initial set of random solutions $\mathbb{CM}$ is generated (line 15). This set serves as antibodies that will be used to counteract the antigens (i.e., the ongoing attack). Besides, also in the dynamic variant, the maturity $\mathcal{M}$ (Equation 2) can be used to refine such an initial generation.

Once the initial generation of random antibodies has been achieved, the affinity of those antibodies is determined (line 17). Specifically, as shown in Algorithm 7 for each asset $A_x \in Y(A_x)$ exploited by an ongoing attack T_k, and for each atomic countermeasure cm_h belonging to the solutions $CM_j \in \mathbb{CM}$, the benefit B of the countermeasures (Equation 10) and the RL (Equation 21) are evaluated. In this direction, if more reaction steps have been selected to be implemented on the same asset A_x, the combined benefit $B(CM_j(A_x))$ must be computed as specified in Equation 15. Further, for each $CM_j \in \mathbb{CM}$, the corresponding fitness function $f(CM_j)$ is computed, as shown in Equation 23. Besides, the average affinity of the different antibodies is calculated.

Then, in line 18, the antibodies with the highest affinity (i.e., set of atomic countermeasures able to minimize the risk) are cloned. This step is equivalent to Algorithm 3 proposed for the static version, where K clones $\{CM_{j_1}, \ldots, CM_{j_K}\}$ are generated proportionally to the fitness function f.

Later, line 19 performs the mutation phase of the clones $\{CM_{j_1}, \ldots, CM_{j_K}\}$, as reported in Algorithm 8. Precisely, each set of atomic countermeasures (i.e., cloned antibody) experiences a mutation process consisting of adding or removing an atomic countermeasure or even modifying its parameters with a certain probability P from a clone CM generating CM'. In this direction, the presented mutation modifies the enforcement parameter of a countermeasure with a probability $P = 0.5$, while the addition or removal are equiprobable with a probability $P = 0.25$. Consequently, the affinity of the mutated clones $f(CM')$ is computed (i.e., the fitness function is measured). If such affinity is greater than the average affinity of the antibodies in the solution space $avg\_affinity(\mathbb{CM})$ (i.e., the atomic countermeasure of the clones are not adequate against the attack), then such

Algorithm 7 `Determine_Affinity`$(T, A, \mathbb{CM})$

```
for all $T_k \in T$ do                                              ▷ For each ongoing attack
    for all assets $A_x \in Y(A_x)$ targeted by $T_k$ do            ▷ For each involved asset
        for all $CM_j \in \mathbb{CM}$ do                           ▷ For each antibody
            for all $cm_h \in CM_j$ do                              ▷ For each atomic countermeasure
                Calculate $B(cm_h)$
            end for
            if $\exists\, CM_j(A_x) \subseteq CM_j$ s.t. $|CM_j(A_x)| > 1$ then   ▷ If two or more atomic cms enforced on the same asset
                Calculate combined benefit $B(CM_j(A_x))$
            end if
            Evaluate the risk $RL(T_k, A_x, CM_j(A_x))$             ▷ Calculate risk of each antibody
        end for
    end for
end for
for all $CM_j \in \mathbb{CM}$ do
    Evaluate $f(CM_j)$                                              ▷ Compute fitness of each antibody
end for
$avg\_affinity(\mathbb{CM}) \leftarrow \sum_{CM_j \in \mathbb{CM}} f(CM_j)/|\mathbb{CM}|$   ▷ Average affinity of the antibodies
```

Algorithm 8 `Mutate_Clones`$(\{CM_{j_1}, \ldots, CM_{j_K}\})$

```
for all K clones $CM \in \{CM_{j_1}, \ldots, CM_{j_K}\}$ do
    $P \leftarrow random(0, 1)$
    if $P > 0.5$ then                                               ▷ With probability $P = 0.5$
        Modify random parameter $p$ in $cm$ within $CM$ to produce $CM'$
    else if $P < 0.25$ then                                         ▷ With probability $P = 0.25$
        Add random $cm$ to $CM$ to produce $CM'$
    else                                                            ▷ With probability $P = 0.25$
        Remove random $cm$ to $CM$ to produce $CM'$
    end if
    Evaluate $f(CM')$                                               ▷ Determine the affinity of the mutated clone $CM'$
    if $f(CM') > avg\_affinity(\mathbb{CM})$ then
        Discard mutated clone $CM'$
    else
        Replace lowest affinity antibody in $\mathbb{CM}$ with $CM'$
    end if
end for
```

clones are removed. Otherwise, they are added to the solutions set $\mathbb{CM}$, replacing the lowest affinity antibodies in $\mathbb{CM}$ at that instant.

Afterward, in line 20, the antibodies with the lowest affinity are removed from the solution space $\mathbb{CM}$. This step of the dynamic reaction is equivalent to Algorithm 5 described for the static version.

Finally, the best antibody is chosen as a final solution (line 21). The loop of lines 16-22 of Algorithm 6 endures till the stop condition is not met. So, the selection of a specific stop condition is vital to perform an acceptable number of iterations. To this extent, and in contrast with the AIS static methodology, the dynamic process is performed in a reactive fashion (i.e., the assets of the system are actively under attack). Thus, it is clear that the algorithm needs to be executed by prioritizing the timing factor. Moreover, the acceptable risk level $\widetilde{RL}$ can also be used to improve the selection of the optimal set of countermeasures.

Once the loop in lines 16-22 ends, and consequently, the best antibody has been found, the atomic countermeasures forming the solution go through an updating stage (lines 23-27). More specifically, like in the static reaction, the links to the external common security knowledge bases are updated, as well as the maturity of the atomic security measures against the eradicated attack. Then, the attack set T is emptied.

C. EXPLANATORY EXAMPLE

To help the reader further understand the proposed methodology, let us illustrate its capabilities and potentialities using an explanatory example. In particular, a Smart Home scenario is presented, where an external attacker is actively trying to

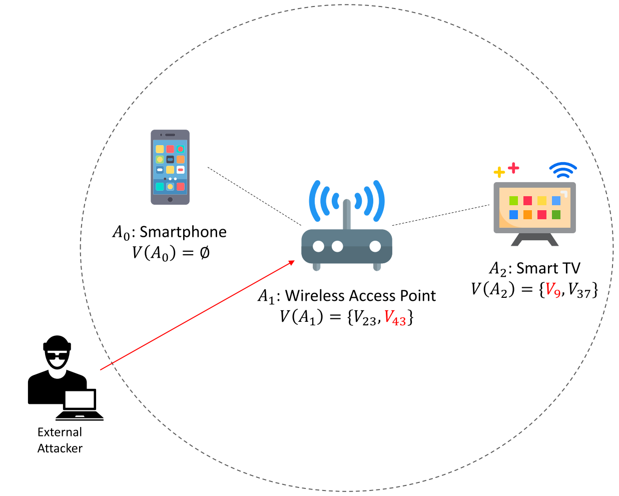

A_0: Smartphone
$V(A_0) = \emptyset$
A_1: Wireless Access Point
$V(A_1) = \{V_{23}, V_{43}\}$
A_2: Smart TV
$V(A_2) = \{V_9, V_{37}\}$
External Attacker

FIGURE 3. Abstract view of the Smart Home scenario proposed for this example.

jeopardize the interconnected smart devices. Figure 3 depicts the mentioned scenario, detailing the assets as follows:

- **Asset A_0**: It is a smartphone connected to a wireless Access Point (AP) to gain Internet access. Such a device does not possess any vulnerability in the proposed scenario, that is, $V(A_0) = \emptyset$.
- **Asset A_1**: It is a Wireless AP providing Internet connectivity to the smart devices. In this example, this device exposes two vulnerabilities, so $V(A_1) = \{V_{23}, V_{43}\}$.
- **Asset A_2**: It is a Smart TV connected to the AP to stream multimedia content. This device exhibits two vulnerabilities, that is, $V(A_2) = \{V_9, V_{37}\}$.

Additionally, in this example, we assume that the external attacker is executing an attack against the AP to gain access to the Smart Home network. Specifically, the malicious actor is exploiting the vulnerability V_{43} of the wireless AP, which allows external attackers to take complete control of the device.[6] Then, the next target of the multi-step attack is represented by the Smart TV that the attacker aims to compromise through the vulnerability V_9. Such a security flaw, in turn, allows remote attackers to cause a Denial of Service (DoS) via a crafted web page.[7] Thus, the attack path of the malevolent external entity contemplates the execution of T_1 to exploit V_{43}, and, once this step is accomplished, the implementation of T_2 to misuse V_9.

In an effort to dynamically shield the targeted smart devices, the AIS dynamic reaction is fired when the attack T_1 is reported. Specifically, the set of assets that this reaction aims to protect are $Y(A_1) = \{A_0, A_1, A_2\}$, but A_0 is excluded since it does not expose vulnerabilities. The parameters involved in the calculation of the optimal set of atomic countermeasures are illustrated in Table 1.

Regarding the wireless AP, it is considered a quite high critical asset within the network, thus $CR(A_1) = 0.85$. Consequently, the acceptable risk level is $\widetilde{RL}(A_1) = 1.5$. In addition, the impact of the exploited vulnerability is $I(T_1) = 8.8$, which is the base value calculated by the CVE. Then, we assume that the probability of exploiting such vulnerability is high, i.e., $P(T_1, A_1) = 0.95$, which can be seen as the confidence that the security devices reporting the detection (e.g., IDS) has. So, the measured risk level in this situation is $RL(T_1, A_1) = 0.95 \times 8.8 \times 0.85 = 7.1$. To counteract the ongoing attack, several dynamic countermeasures are listed in Table 1, together with their residual values. Recall that such values become real when the atomic countermeasures are actually enforced.

Regarding the Smart TV, its criticality value is $CR(A_2) = 0.7$, generating an acceptable risk level $\widetilde{RL}(A_2) = 3$. Moreover, the impact that the violation of the exposed vulnerability may cause is $I(T_2) = 7.5$, which, also in this case, is the base value computed by the CVE. Additionally, the probability of exploiting such vulnerability is $P(T_2, A_2) = 0.7$, which we assume is the value that the attack model has reported for this scenario. Consequently, the risk level for this asset is $RL(T_2, A_2) = 3.67$. To block the possible steps of the attackers, the countermeasures in Table 1 are proposed, and, similarly to the previous case, their values are residual.

Yet, some considerations should be made to contextualize the example thoroughly. First, it is worth mentioning that the presented countermeasures are just a few potential remediations that can be enforced on the threatened assets to eradicate the attack. Furthermore, it has to be noted that it is not realistic to expect the certain presence of a SIEM solution in this scenario. However, some lightweight alternatives can be implemented to preserve the security level of the smart devices, which are becoming more important every day and critical within the scenario in which they are deployed [2]. Also, the number of possible combinations of atomic countermeasures to be enforced is relatively high, even in such a small example. In fact, considering only the possible activations of a countermeasure (without weighing the parametric countermeasures), the number of possible solutions for 11 countermeasures is $211 = 2048$.

The steps of the AIS dynamic reaction for this tiny example are illustrated in Figure 4. First, an initial set of random antibodies is generated. In particular, three antibodies are initially created, i.e., $\mathbb{CM} = \{CM_1, CM_2, CM_3\}$, selecting random atomic countermeasures for their composition, i.e., $CM_1 = \{cm_2, cm_7\}$, $CM_1 = \{cm_5, cm_6\}$, and $CM_3 = \{cm_1, cm_8\}$. Later, the affinity of the generated antibodies must be calculated, as indicated in Algorithm 7. At this point, those atomic security measures assume their real value against the ongoing threat, which are reported in Table 2. From those values, it is possible to calculate the benefit $B(cm_i)$ (Equation 10). Since no multiple countermeasures have been selected to be executed on the same asset, no combined effect $B(\{cm_i, cm_j, \ldots\})$ needs to be calculated in this case. Then, the risk level $RL(T_k, A_x, cm_i)$ is calculated again considering the benefit provided by implementing the selected atomic countermeasures. Additionally, the fitness of each antibody is calculated (Equation 23), and, subsequently, their average

[6] https://nvd.nist.gov/vuln/detail/CVE-2017-13772
[7] https://nvd.nist.gov/vuln/detail/CVE-2019-11889

TABLE 1. AIS dynamic example settings for the Smart Home scenario.

Asset A_x	$CR(A_x)$	$\widetilde{RL}(A_x)$	Attack				Countermeasure					
			V_i	$I(T_k)$	$P(T_k, A_x)$	$RL(T_k, A_x)$	ID	Eff	Imp	$Cost$	P	δ
Wireless AP A_1	0.85	1.5	V_{43}	8.8	0.95	7.1	cm_1	L	L	L	# seconds	• Block external IP address • SW – short-term • Dynamic
							cm_2	H	M	L	# seconds	• Block AP port • SW – short-term • Dynamic
							cm_3	H	H	M	# seconds or # frames or # alerts	• AP isolation • SW – short-term • Dynamic
							cm_4	H	L	L	# version	• Update AP Firware • SW – long-term • Dynamic
							cm_5	M	M	M	authentication factor	• Additional authentication • SW – long-term • Dynamic
Smart TV A_2	0.7	3	V_9	7.5	0.7	3.67	cm_6	M	M	L	# seconds	• Block internal IP address • SW – short-term • Dynamic
							cm_7	H	H	M	# seconds or # frames or # alerts	• Host isolation • SW – short-term • Dynamic
							cm_8	H	L	L	N.A.	• Patch vulnerability • SW – long-term • Dynamic
							cm_9	H	H	L	# seconds	• Unplug Smart TV • HW – short-term • Dynamic
							cm_{10}	H	X	H	N.A.	• Change Smart TV • HW – long-term • Static
							cm_X	X	X	X	N.A.	• No action

TABLE 2. AIS dynamic countermeasures for the Smart Home scenario.

Antibody	ID	$Eff(cm_i)$	$Imp(cm_i)$	$Cost(cm_i)$	$B(cm_i)$	$RL(T_k, A_x, cm_i)$	$f(CM_i)$
CM_1	cm_2	0.7	0.4	0.2	4.62	1.53	0.985
	cm_7	0.8	0.7	0.4	3.43	1.06	
CM_2	cm_5	0.6	0.5	0.5	3.32	2.13	1.37
	cm_6	0.5	0.3	0.2	4.08	0.89	
CM_3	cm_1	0.3	0.1	0.2	3.32	2.13	1.53
	cm_8	0.9	0.2	0.2	6.33	0.57	

affinity $avg\_affinity(\mathbb{CM}) = (0.985 + 1.37 + 1.53)/3 = 1.348$.

Therefore, the antibodies must pass through the cloning and mutation phases. For the sake of simplicity, this example is run selecting $K = 1$. Thus, the best antibody CM_1 is cloned. To this extent, we assume that the probabilistic mutation phase in Algorithm 8 selects to modify the enforcement parameter of $cm_2 \in CM_1$ generating CM_1'. Specifically, such modification, say, increases the number of seconds during which the AP port is blocked to stop the external attacker. In this case, the fitness value of the mutated antibody CM_1' is $f(CM_1') = 0.98$, slightly improving $f(CM_1)$, since the benefit of the mutated countermeasure is $cm_2' = 4.79$. Given that the affinity of CM_1' is lower than the average affinity $avg\_affinity(\mathbb{CM})$, the mutated antibody becomes part of the solution space $\mathbb{CM}$ replacing CM_3 (i.e., the worst affinity antibody at this step). Finally, the worst antibody within the solution space CM_2 is removed, and a new randomly generated one replaces it, say, CM_4, to start a new iteration of the AIS methodology.

V. EXPERIMENTS

Several thoughtful experiments have been conducted to demonstrate the capabilities and feasibility of the proposed methodology. Specifically, the tests have been executed on a Toshiba Portege Z30-C laptop equipped with an Intel Core i7-6500U CPU and 16 GB of DDR4 memory.

For ease of readability, the settings of the experiments are reported in Section V-A, a thorough analysis of the obtained

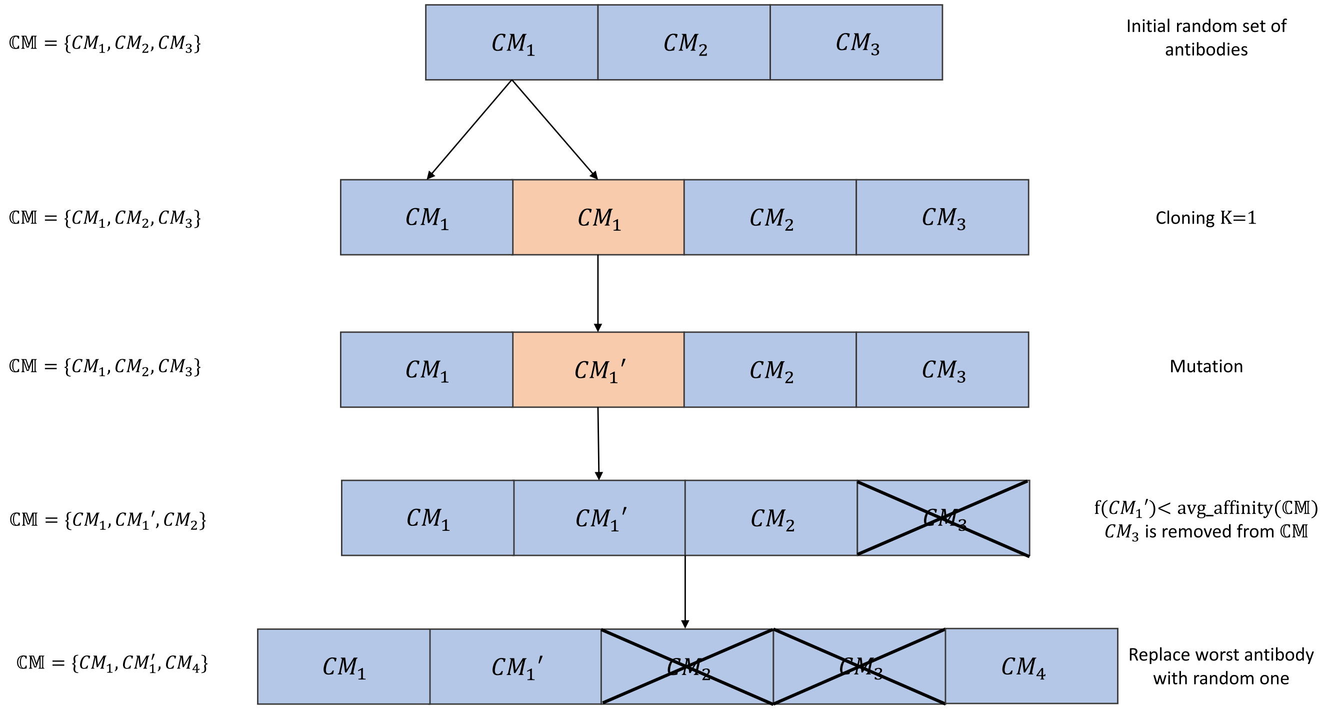

$\mathbb{CM} = \{CM_1, CM_2, CM_3\}$
CM_1
CM_2
CM_3
Initial random set of antibodies
$\mathbb{CM} = \{CM_1, CM_2, CM_3\}$
CM_1
CM_1
CM_2
CM_3
Cloning K=1
$\mathbb{CM} = \{CM_1, CM_2, CM_3\}$
CM_1
CM_1'
CM_2
CM_3
Mutation
$\mathbb{CM} = \{CM_1, CM_1', CM_2\}$
CM_1
CM_1'
CM_2
CM_3
f(CM_1')< avg_affinity($\mathbb{CM}$)
CM_3 is removed from $\mathbb{CM}$
$\mathbb{CM} = \{CM_1, CM_1', CM_4\}$
CM_1
CM_1'
CM_2
CM_3
CM_4
Replace worst antibody with random one

FIGURE 4. Example: AIS dynamic reaction steps for the presented example.

TABLE 3. Description and values for each input parameter used in the experiments.

Parameter	Description	Value	Domain
$\|\tau\|$	Number of threats within the protected system	$[20, 100]$	$\mathbb{N}$
$P(\tau_k, A_x)$	Probability of the occurrence of the threat τ_k over the asset A_x	`RAND(0,1)`	$\mathbb{R}$
$I(\tau_k)$	Impact of the threat τ_k over the asset A_x	`RAND(0,10)`	$\mathbb{R}$
N_A	Number of assets belonging to the protected system	$[20, 100]$	$\mathbb{N}$
$CR(A_x)$	Criticality of the asset A_x	`RAND(0,1)`	$\mathbb{R}$
$\|CM\|$	Number of countermeasures to be enforced	$[100, 1{,}000]$	$\mathbb{N}$
$Eff(cm_i)$	Effectiveness of the countermeasure cm_i	`RAND(0,1)`	$\mathbb{R}$
$Imp(cm_i)$	Negative impact of the countermeasure cm_i	`RAND(0,1)`	$\mathbb{R}$
$Cost(cm_i)$	Cost of the countermeasure cm_i	`RAND(0,1)`	$\mathbb{R}$
$\|\mathbb{CM}\|$	Number of antibodies	$[10, 40]$	$\mathbb{N}$
K	Number of clones	$\|\mathbb{CM}\|/3$	$\mathbb{N}$
$Iter$	Number of iterations of the AIS-powered reaction	$[100, 1{,}000]$	$\mathbb{N}$

results is detailed in Section V-B, while a discussion on the limitation of the approach are presented in Section V-C.

A. SETTINGS

The AIS-powered reaction methodology has been implemented from scratch as a group of interconnected Python scripts. During the experimental sessions, the scripts were running on separate CPUs in order to avoid conflicts as a consequence of possible context switches.

Due to the inherent characteristics of the methodology (as detailed in Section III), several parameters need to be considered and tuned to achieve an optimal configuration and, subsequently, react against potential threats effectively. In particular, Table 3 resumes the input parameters to be set, together with the values used. To this extent, it has to be noted that several values (i.e., $P(\tau_k, A_x)$, $I(\tau_k)$, $CR(A_x)$, $Eff(cm_i)$, $Imp(cm_i)$, and $Cost(cm_i)$) are generated randomly to better argue on the capabilities of the presented methodology by introducing randomness.

Additionally, the random generation of the parameters of the threat (i.e., $P(\tau_k, A_x)$ and $I(\tau_k)$) allows us to generalize the reaction efficiency of the AIS-powered methodology since the experiments do not focus on specific attacks or suspicious activities. In this sense, the AIS-powered methodology is agnostic to the specific threat, as long as it is correctly detected and reported to the SIEM. From those reports,

TABLE 4. Description of the output parameters measured during the experiments.

Parameter	Description
$f(CM_i)$	Fitness value of the best solution
Execution time	Execution time of the algorithm (in seconds)

the SIEM is in charge of extracting the relevant parameters that will fire the reaction.

Additionally, some of the input parameters related to the system under protection are chosen within an assigned range, for instance, N_A, $|\tau| \in [20, 100]$, $|CM| \in [100, 1{,}000]$. This choice relies on the fact that we believe that such configurations represent realistic environments of modern systems without loss of generality. Besides, the output parameters measured during the experiments are illustrated in Table 4. Particularly, it is worth remarking that the possible fitness values are included in [0, 10], where 0 indicates the optimal solution, and 10 refers to the worst one.

Before executing the actual experimental sessions, various tests have been executed, aiming at acquiring an initial knowledge on the sensitivity of the parameters and how they affect the overall performance of the algorithm. The results of such tests are reported in Figure 5. Those tests have been carried out by defining a fixed number of assets (i.e., $N_A = 20$) and threats (i.e., $\tau = 20$) to simulate a network in which 20 compromised assets are operating, which represents a small system without loss of generality. In particular, for each row of Figure 5, two parameters are fixed, and the third one is changed to discuss its impact on the fitness and execution time. It has to be stressed out that each experiment has been repeated 100 times to avoid potential outliers due to the randomness of the considered entities. Additionally, the first Y-axis (i.e., concerning the fitness) has been limited to the [0, 1] interval, whereas the secondary Y-axis (i.e., relative to the execution time) has been plotted on a logarithmic scale.

Going into detail, the first row (Figure 5a-5b-5c) measures the impact of an increasing number of antibodies within the solution space, fixing $Iter = 1{,}000$ and $|CM| = 100$. Concretely, it is possible to observe that an increasing count of antibodies improves the fitness slightly while deteriorating the execution time since the algorithm needs to manage a wider population.

Next, the second row (Figure 5d-5e-5f) illustrates the consequence of a growing quantity of countermeasures with $Iter = 100$ and $|\mathbb{CM}| = 20$. Particularly, the fitness value ameliorates slightly, whereas the execution time worsens as the methodology has to calculate more benefit values.

Then, the third row (Figure 5g-5h-5i) depicts the effect of a larger number of iterations setting $|CM| = 400$ and $|\mathbb{CM}| = 10$. Specifically, the fitness enhances more significantly in this case, but the algorithm runs for a longer time.

Finally, the fourth and last row is composed of a unique graph (i.e., Figure 5j), in which the inspected parameters are boosted to the maximum value of the current experiment simultaneously, i.e., $Iter = 1{,}000$, $|CM| = 1{,}000$, $|\mathbb{CM}| = 30$. As expected, the fitness value reaches its lowest (thus best) value, but the AIS-reaction execution becomes relatively slow.

All in all, this experimental phase allows one to conclude that a higher number of iterations or antibodies harms the timing performance of the methodology, which overall operates satisfactorily given the reduction of the fitness value in each test. On the contrary, an increasing number of countermeasures slightly degrades the execution time of the algorithm, as expected.

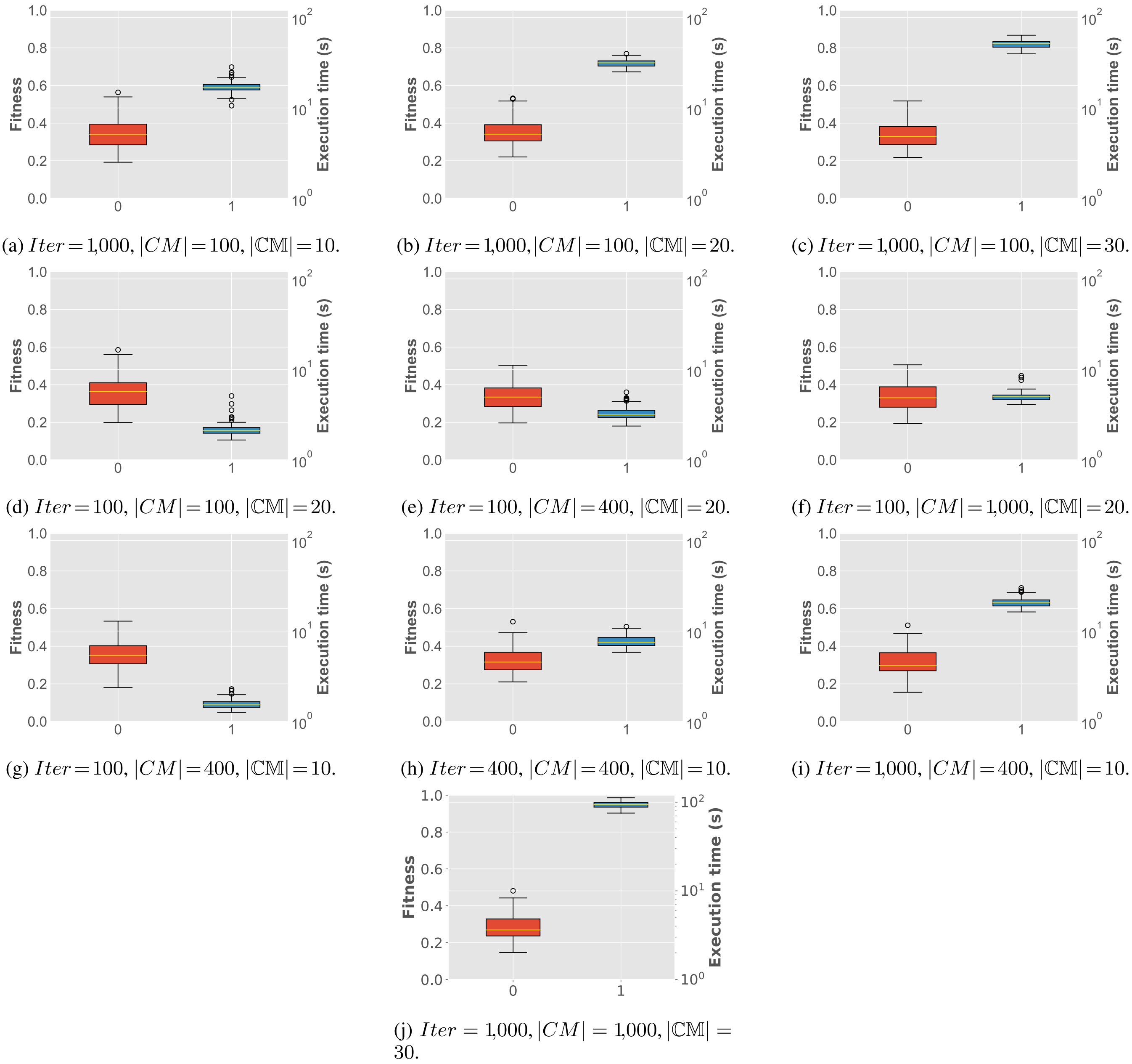
B. ANALYSIS OF RESULTS

This section offers an in-depth analysis and discussion on the outcomes of the several experiments conducted on the proposed AIS-powered methodology. Note that each of the presented experiments has been repeated 100 times to avoid the effects of potential outliers due to the randomness of the used parameters, as reported in Table 3.

1) ON THE PERFORMANCE OF THE AIS-POWERED REACTION

First, the fitness values and execution time have been measured while increasing the number of iterations, as illustrated in Figure 6. For this test, $N_A = 20$ and $\tau = 20$ are randomly generated, simulating the existence of 20 compromised assets within the protected system. In addition, the number of countermeasures and antibodies is fixed, i.e., $|CM| = 1{,}000$ and $|\mathbb{CM}| = 20$. Explicitly, at each iteration, the output parameters are measured and, afterward, the arithmetic medians over 100 repetitions are plotted (i.e., the dashed blue line for the fitness and the dashed-dotted green line for the execution time, respectively) together with the corresponding standard deviations (the red vertical line and the black one, respectively). Thus, the plot in Figure 6 portrays two trends of a thousand points along with the relative standard deviations.

Regarding the fitness curve, it starts from an initial value representing the system without any countermeasure enforced. Next, the AIS algorithm initiates, computing the solution that aims at minimizing the fitness. While the iterations number increases, the fitness of the best solution enhances, stabilizing its value after 200 iterations approximately. The improvement is then quite contracted down to the end of the experiment, ending in $f \approx 0.3$. To this extent, it should be pointed out that, since the values of the countermeasures, assets, and threats parameters are generated randomly, it is very improbable that a solution exists (i.e., a combination of atomic countermeasures) minimizing the



(a) $Iter = 1{,}000$, $|CM| = 100$, $|\mathbb{CM}| = 10$.

(b) $Iter = 1{,}000, |CM| = 100$, $|\mathbb{CM}| = 20$.

(c) $Iter = 1{,}000$, $|CM| = 100$, $|\mathbb{CM}| = 30$.

(d) $Iter = 100$, $|CM| = 100$, $|\mathbb{CM}| = 20$.

(e) $Iter = 100$, $|CM| = 400$, $|\mathbb{CM}| = 20$.

(f) $Iter = 100$, $|CM| = 1{,}000$, $|\mathbb{CM}| = 20$.

(g) $Iter = 100$, $|CM| = 400$, $|\mathbb{CM}| = 10$.

(h) $Iter = 400$, $|CM| = 400$, $|\mathbb{CM}| = 10$.

(i) $Iter = 1{,}000$, $|CM| = 400$, $|\mathbb{CM}| = 10$.

(j) $Iter = 1{,}000, |CM| = 1{,}000, |\mathbb{CM}| = 30$.

FIGURE 5. Fitness and execution time values for the initial experiments.

fitness till extremely small values unless the experiments are executed for a huge number of times.

Concerning the execution time trend, it increases almost linearly with the number of executions until 60 seconds on average for $Iter = 1{,}000$, expanding the standard deviation values when the iterations number gets bigger. Considering that the fitness stabilizes around 200-400 iterations, the algorithm requires between 10 and 20 seconds to calculate the best solution, which can be assumed acceptable in the proposed scenario.

Moreover, the output parameters have been determined while increasing the number of countermeasures from 100 to 1,000 with increments of 100, as shown in Figure 7. Also for this test, $N_A = 20$ and $\tau = 20$ are randomly generated. Besides, the number of iterations and antibodies are set, i.e., $Iter = 200$, $|\mathbb{CM}| = 20$. Specifically, the arithmetical medians and standard deviations over 100 runs of the best solution fitness and execution time are charted for each run of the experiments. Successively, the medians of the curves are connected using the same shapes and colors of the previous experiment.

About the fitness values, a higher number of atomic countermeasure does not provide a substantial improvement. That is, the fitness slightly improves as the countermeasures

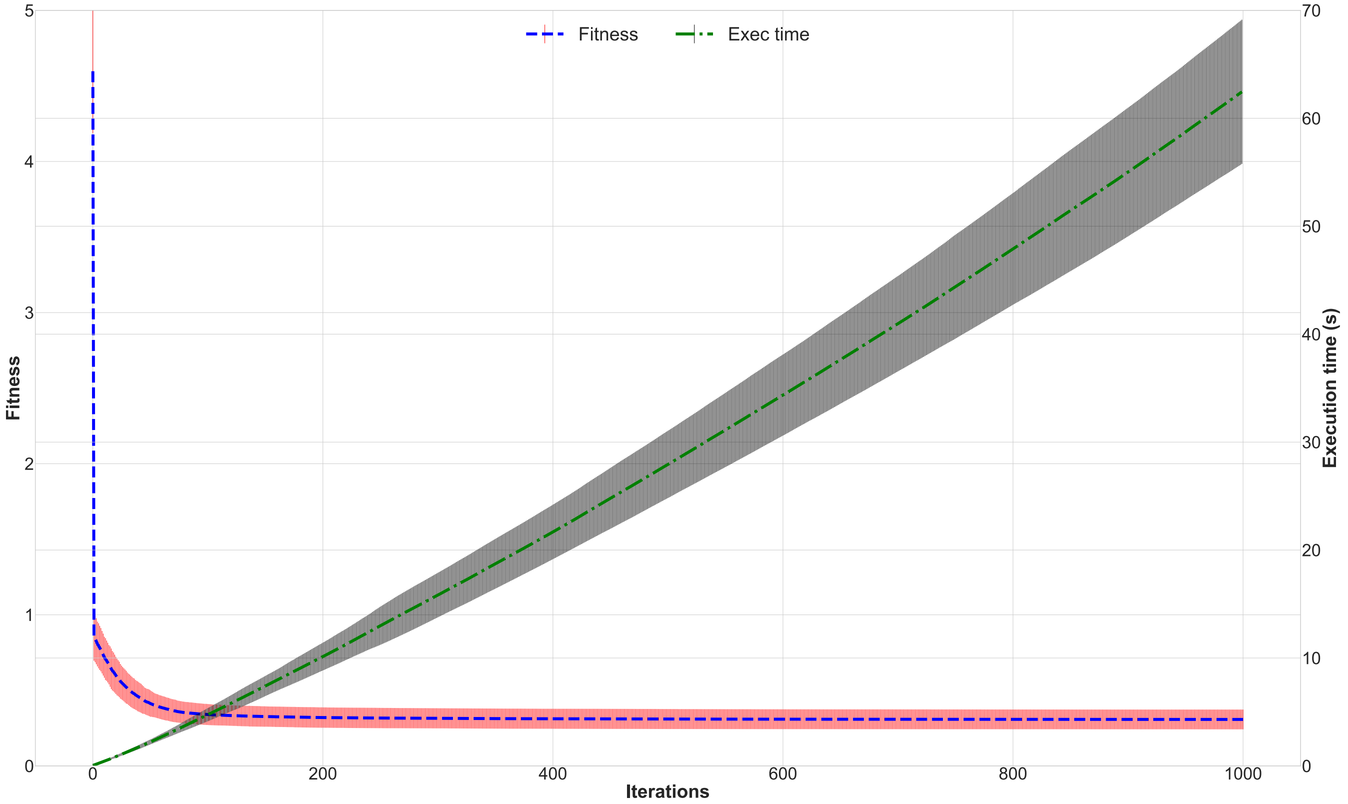


FIGURE 6. Fitness and execution time while increasing *Iter*, $|CM| = 1{,}000$, $|\mathbb{CM}| = 20$.

augment, as already proved during the initial experimental phase. Nevertheless, the best solution offers a remarkable fitness value, i.e., $f \approx 0.3 - 0.35$.

Instead, the execution time presents a less-than-linear increasing trend, since the algorithm concludes its execution between 4.5 (in the case of $|CM| = 100$) and 10.5 seconds (for $|CM| = 1{,}000$) on average. Therefore, the number of countermeasures does not imply a significant negative impact on the execution time of the AIS reaction methodology.

Furthermore, the fitness value and execution time have been evaluated while increasing the number of antibodies from 10 to 40 with increments of 5, as depicted in Figure 8. Also for this experiment, $N_A = 20$ and $\tau = 20$ are randomly generated. In addition, the number of countermeasures and iterations are fixed, i.e., $|CM| = 400$, $Iter = 200$. In particular, the arithmetical medians and standard deviations over 100 runs of the best solution fitness and execution time are drawn for each run, and, afterward, the medians of the curves are connected using the same shapes and colors of the previous experiments.

With regard to the fitness curve, the number of antibodies (i.e., possible solutions) does not produce a considerable enhancement. Explicitly, the fitness of the best solution slightly ameliorates as the population size grows, as already argued during the initial experiments. However, the computed fitness for the best individual is quite adequate, i.e., $f \approx 0.25 - 0.35$. Recall that a wider population size implies a bigger number of clones directly since the latter is calculated based on the former.

Concerning the execution time, the trend is almost linear with the number of antibodies. That is, the algorithm terminates its execution between 3.5 (for $|\mathbb{CM}| = 10$) and 12.2 seconds (considering $|\mathbb{CM}| = 40$) on average.

In order to argue on the scalability of the proposed methodology, an experiment has been executed to measure the output parameters with an increasing number of compromised assets from 10 to 100 with increments of 10, as exposed in Figure 9. Similarly, the total number of atomic countermeasures is augmented, i.e., $|CM| = 10 \times N_A$. Also, the iterations and antibodies are set, i.e., $Iter = 200$, $|\mathbb{CM}| = 20$. Also in this case, the arithmetical medians and standard deviations over 100 runs of the best solution fitness and execution time are plotted for each run, and, later, the medians of the curves are connected using the same shapes and colors of the previous experiments.

Specifically, the fitness is not negatively impacted by the number of compromised assets. In fact, its median value is close to 0.3 on average for 10 assets and increases till reaching 0.5 for 100 assets. One could easily say that such a rise is more than acceptable considering the magnitude of the considered scenario.

The execution time, instead, presents an almost-linear tendency compared to the number of assets. Nonetheless, very few will oppose that a methodology that is able to provide

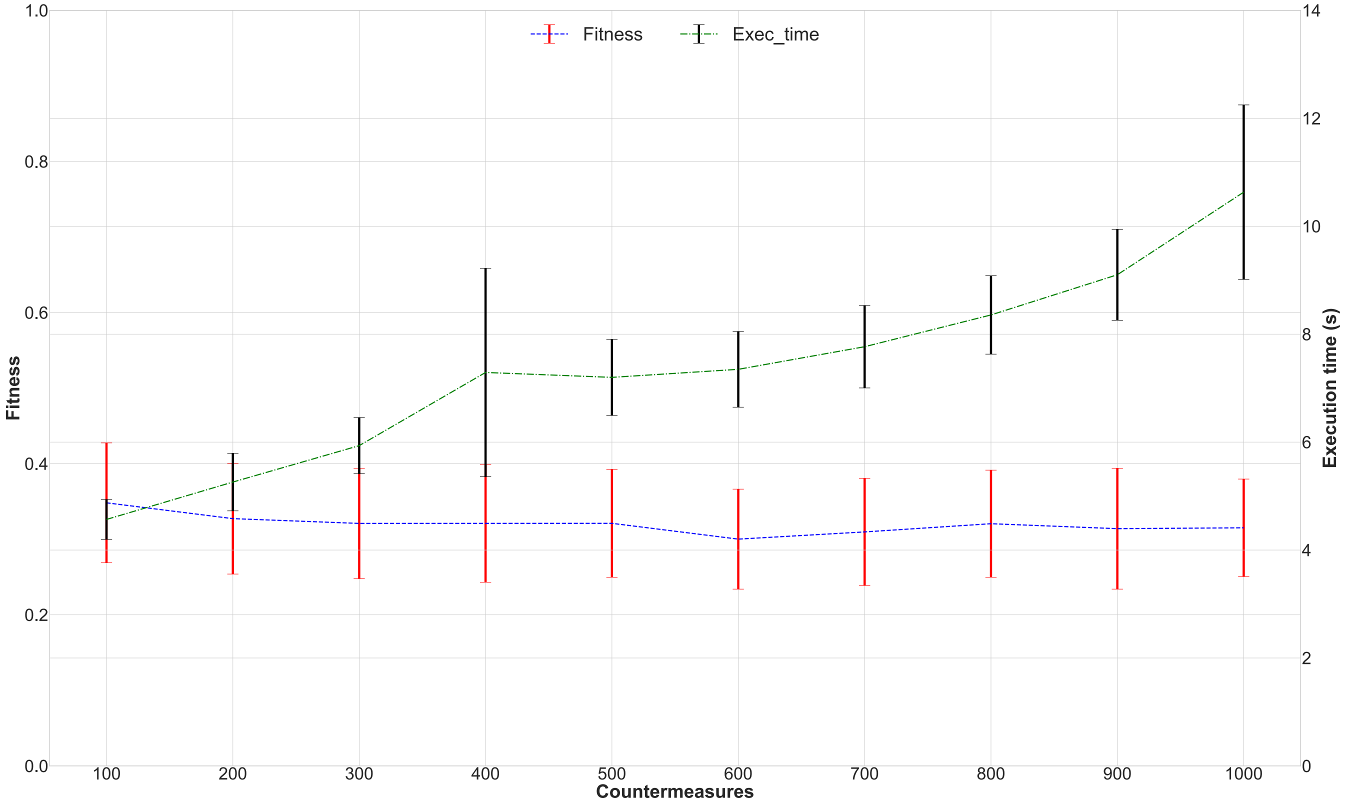


FIGURE 7. Fitness and execution time while increasing $|CM|$, $Iter = 200$, $|\mathbb{CM}| = 20$.

countermeasures for 100 assets in 60 seconds represents an outstanding result.

2) ON THE IMPORTANCE OF THE STOP CONDITION

As previously mentioned in Section IV, an accurate choice of the stop condition is essential for each proposed methodology of reaction frameworks. Particularly, an erroneous assignment of this parameter could expose the protected system to dangerous situations or in an over-utilization of the system resources. Generally, the stop condition can be categorized as reported in the following list:

- **Time-based stop condition**: The algorithm terminates after a certain predefined lapse of time.
- **Iteration-based stop condition**: The algorithm concludes when a finite number of iterations is achieved.
- **Quality-based stop condition**: The algorithm stops when the quality of the solution reaches a predefined threshold.
- **Risk-based stop condition**: The algorithm ends if the risk has been minimized to a fixed value.
- **Context-based stop condition**: The algorithm completes when a parameter or entity moves to a specific state.

It has to be stated that it is also possible to combine the categories above in order to build a fine-grained stop condition that better fits the particular needs of a certain algorithm.

Particularly in our research, a context-aware stop condition has been developed and tested to argue on its capabilities within the proposed AIS-reaction methodology. Concretely, such a stop condition is calculated *on-the-fly* based on the fitness value witnessed in the system. That is, the observed fitness serves as a security index to determine the most appropriate stop condition at each execution. In this sense, we assume that the higher (worse) the fitness value is measured, the faster the reaction needs to be enforced on the assets. The main reasoning behind this choice lies in the fact that, when the fitness reaches high values (i.e., close to 10), the assets of the protected system are exposed to an extremely high risk demanding a fast response. On the contrary, if the fitness remains at low rates (i.e., around 0), the reaction may be more time-consuming in search of the optimal combination of countermeasures.

As seen in the former experiments, the selection of input parameters directly impacts the performance of the algorithm, in particular, both on the fitness and execution time. Thus, starting from calculating the initial fitness (when no countermeasures have been enforced), the methodology assigns specific values to the input parameters. Explicitly, for each analyzed parameter, min-max intervals have been determined based on the outcomes of the experiments:

$$\begin{aligned} Iter &\in [100, 975] \\ |CM| &\in [100, 1{,}100] \\ |\mathbb{CM}| &\in [10, 35] \end{aligned} \tag{24}$$

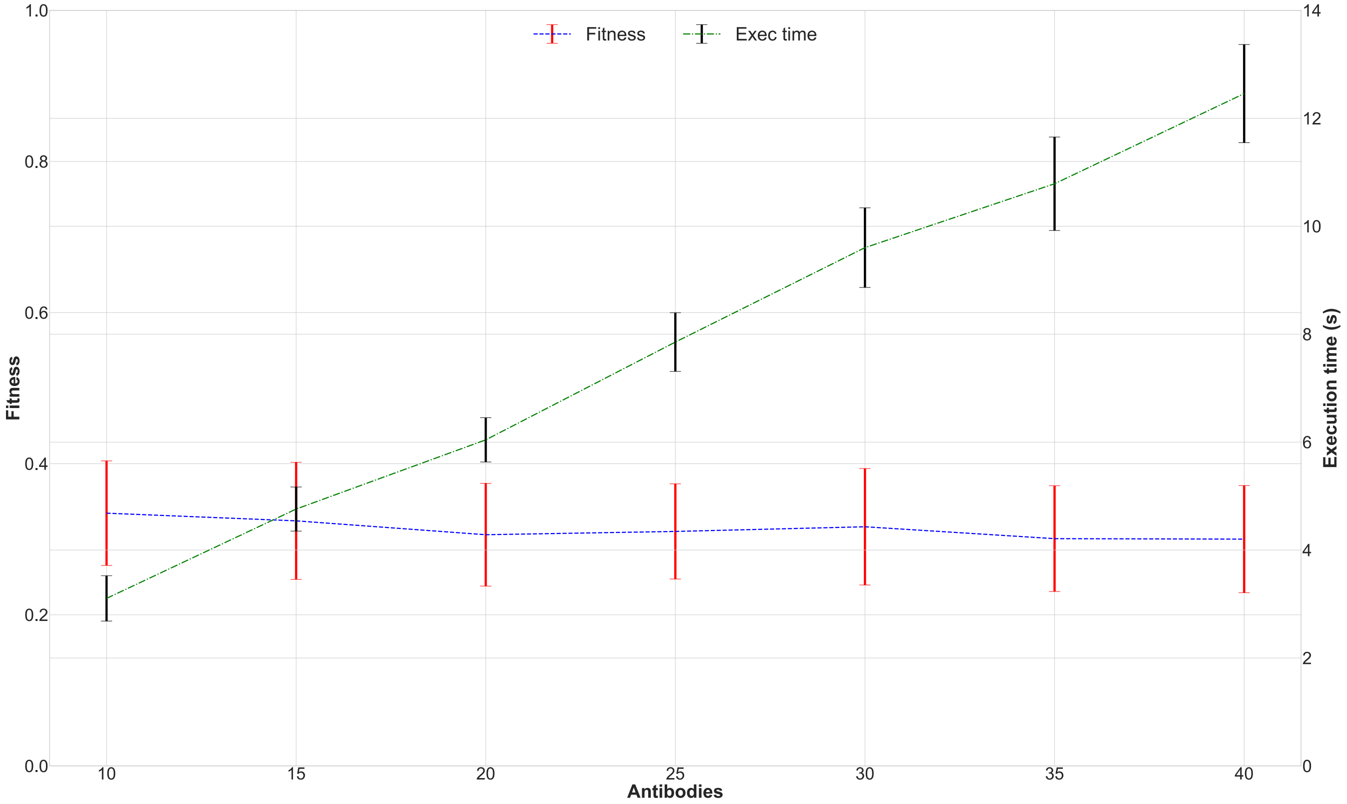


FIGURE 8. **Fitness and execution time while increasing $|\mathbb{CM}|$, $|CM| = 400$, $Iter = 200$.**

Thus, the number of iterations of the algorithm encompasses a range where 100 represents the minimum quantity and 975 the maximum one. Therefore, when the fitness value is computed, the exact number to be assigned to the input parameters $P_i \in \{Iter, |CM|, |\mathbb{CM}|\}$ is calculated as follows:

$$P_i = min_i + (max_i - min_i) \times (1 - f/10) \tag{25}$$

As a matter of example, suppose that the fitness value measured in a specific state of the protected system is equal to 8. Subsequently, the number of iteration with which the algorithm is executed is set to $Iter = 100 + (975 - 100) \times (1 - 8/10) = 275$. Similarly, the same reasoning applies to the other input parameters ($|CM| = 100 + (1,100 - 100) \times (1 - 8/10) = 300$ and $|\mathbb{CM}| = 10 + (35 - 10) \times (1 - 8/10) = 15$).

The results of the experiment employing the context-aware stop condition are shown in Figure 10. In the chart, the Y-axis represents the measured value of fitness during the execution of the algorithm, whereas the X-axis shows the execution time (in seconds) on a logarithmic scale. Specifically, each line of Figure 10 depicts a different situation in which, based on the witnessed initial fitness value, the other parameters are consequently set before the algorithm initiates. Note that the experiment has been executed 100 times, and the plot depicts the arithmetic medians and standard deviations computed over the runs.

Going into detail, it is possible to appreciate that the proposed stop condition performs as expected for the algorithm. In particular, the first curve starts from a fitness value equal to 10 (meaning the worst possible risk situation). Therefore, the other parameters are assigned based on Equation 25: $Iter = 100$, $|CM| = 100$, and $|\mathbb{CM}| = 10$, respectively. In accordance with such parameters, the algorithm runs for 1 second approximately, minimizing the fitness to 1.7 on average by selecting the combination of atomic countermeasure in a reduced time window. On the contrary, the other curves portrayed in the plot begin from lower fitness values (i.e., $f = 8$, $f = 6$, $f = 4$, and $f = 2$), being able to set the input parameters to higher values. Consequently, the algorithm can execute over longer runs, minimizing the fitness till values close to 0 in the best case. That is, when the fitness is equal to 2, the methodology runs for about 100 seconds and, during this time, it is able to reduce the fitness to approximately 0.2 on average.

C. DISCUSSION

Throughout the previous Sections, the AIS-powered reaction has been proposed, featuring the crucial characteristic of the AIS methodology and leveraging both the standard countermeasures representation and the countermeasure benefit. In this direction, the AIS-powered reaction has been proved robust and efficient, being able to compute the optimal set of countermeasures to enforce on the assets in a more than acceptable time while facing several random threat scenarios.

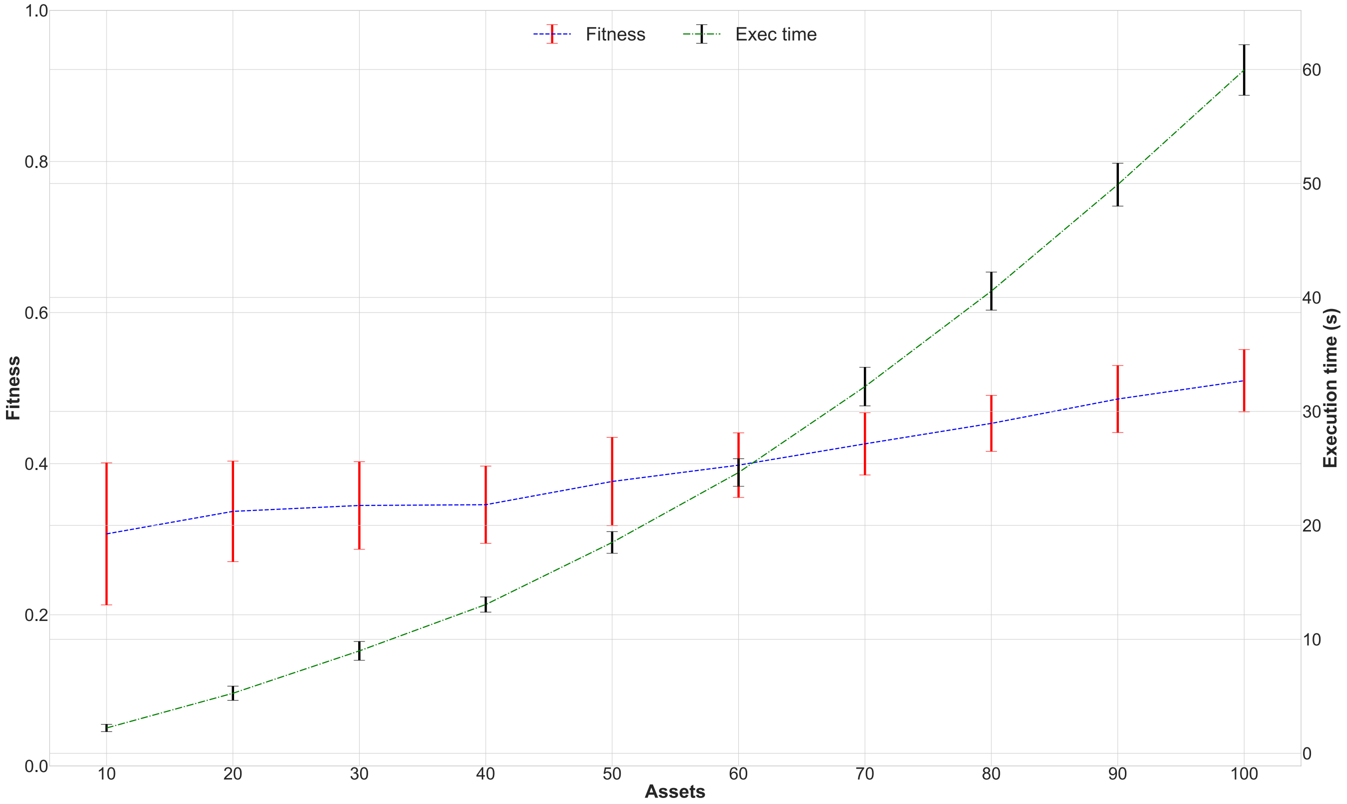


FIGURE 9. **Fitness and execution time increasing N_A, $|CM| = 10 \times N_A$, $Iter = 200$, $|\mathbb{CM}| = 20$.**

Nevertheless, some challenges still remain partially solved or even unsolved.

Firstly, the proposed reaction assumes that correct detection and reporting phases are accurately performed within the protected system. That is, vulnerability scanners (for the static response) and IDSs (for the dynamic version) are in charge of discovering possible suspicious activities and report those in a timely fashion. From such reports, the SIEM extracts relevant parameters about the threat and, consequently, fires the AIS-powered reaction. However, it is licit to assume that those hypotheses may result incorrect in certain circumstances, e.g., the exploitation of a zero-day vulnerability. In this scenario, further investigation is needed to provide the detection entities with powerful attack models that contemplate the potential appearance of such previously unknown activity.

Then, the performance of the AIS-powered reaction has been tested using random inputs, repeating the experiments several times to prove its capabilities. One could say that the outcomes were satisfactory even for a quite dense network, say, with a hundred assets and a thousand countermeasures. Nonetheless, the proposed reaction should be tested in a more realistic environment (e.g., a controlled virtual network) to further discuss potential advantages and possible limitations. Such a research path represents a fascinating future research direction on which we are currently working.

Besides, the input parameters for the experiments have been chosen undertaking reasonable assumptions, for example, considering the size of the protected network or a fair number of countermeasures for each asset. To this extent, the application of an optimization algorithm would be beneficial to pinpoint correct values for those parameters, improving both the security and timing performance of the AIS methodology ultimately.

Another critical challenge regarding the reaction field is the population of the countermeasure knowledge. One could quickly notice that the effort required to study, analyze, and acquire such knowledge about the remediations is not trivial. This limitation is principally due to the often ambiguous definition of countermeasure within the literature. In this sense, we believe that the proposal of a reaction methodology that actively uses a standard representation of countermeasures objects constitutes a fundamental step toward the acquisition of reaction knowledge and its sharing among different security teams.

Last but not least, a crucial point worth discussing is the role of the security administrators. Indeed, they play a primary function in the battle against cyberattacks, balancing the trade-off between the effectiveness and the cost of the reaction. In our vision, each phase of the cybersecurity cycle should not overlook security administrators' feedback, avoiding overwhelming them at any time. For this reason, our proposal of semi-automatic reaction envisions the security

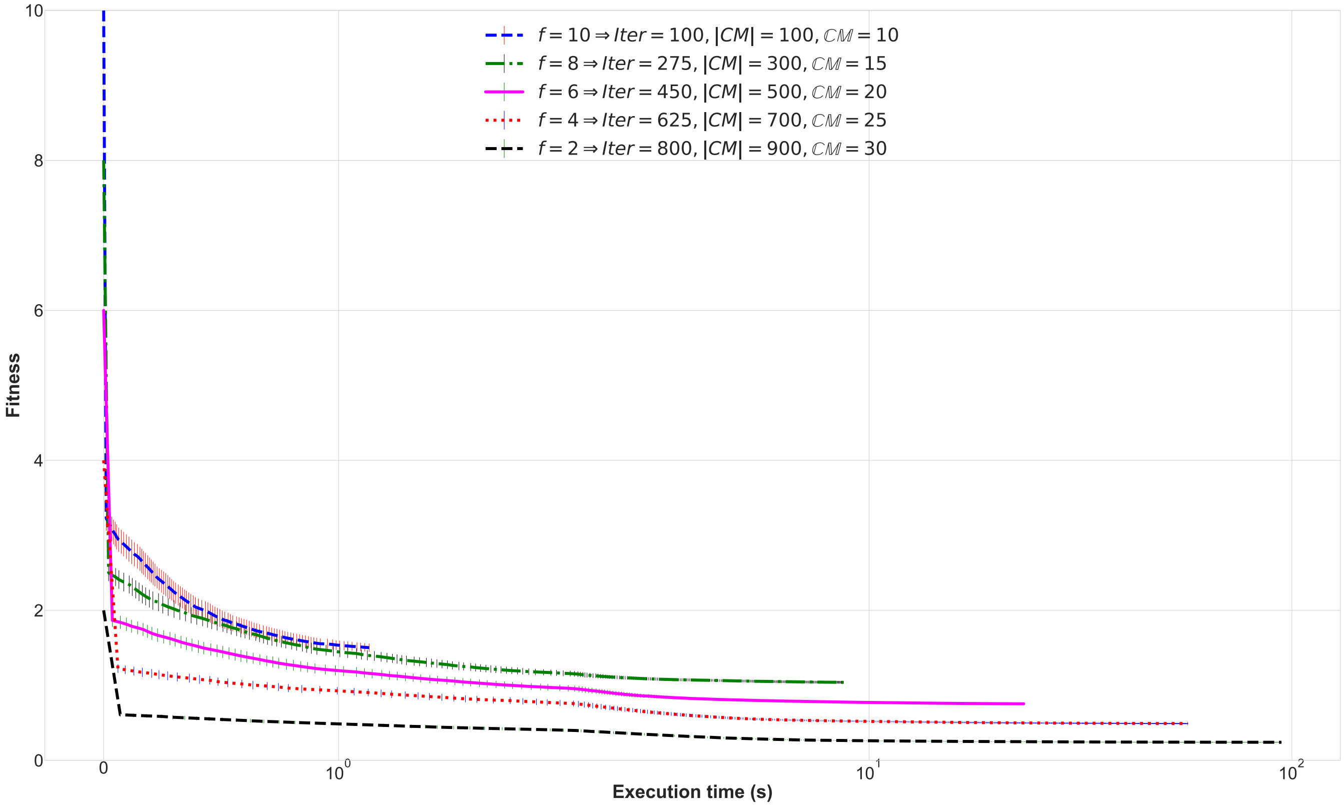


FIGURE 10. Context-aware stop condition trend.

administrators as a central element since they are in charge of selecting the most appropriate countermeasures among the ones computed by the AIS-powered reaction and, consequently, update the maturity of such a bunch of remediations.

VI. CONCLUSION AND FUTURE WORK

It is without a doubt that the recent convergence of ICT technologies around the Internet presents several outstanding advantages but also poses major challenges from a security perspective. Indeed, those infrastructures are suffering a considerable number of cyberattacks that become every day more sophisticated and disruptive, causing huge economic losses. In this endless battle between security teams and malevolent entities, reaction strategies are essential to counteract potential devastating threats. In particular, the optimal set of countermeasures must be cherry-picked to balance the trade-off between the effectiveness of the reaction in eradicating the threat and its possible negative impact on the system properties. Additionally, the selection needs to be adaptable to the identified threat, witnessing the risk at any time.

In this paper, an AIS-powered methodology to select the optimal set of atomic countermeasures is proposed. Specifically, an adaptation of such a bio-inspired technique is presented, modeling the various entities participating in the reaction battlefield to translate them into the immunological knowledge sphere. That is, the detected threats represent the antigens, while the countermeasures serve as the antibodies. Moreover, an index to evaluate the convenience of enforcing a specific atomic countermeasure quantitatively is proposed, i.e., the countermeasure benefit. This index is then extended to embrace the enforcement of multiple countermeasures on a certain asset, adopting a defensive perspective. The AIS-powered reaction is then presented, detailing with fine-granularity the steps needed to achieve the optimal countermeasures selection and distinguishing between static and dynamic AIS-reactions. To this extent, the studied reactions do not aim to reduce the risk blindly but to minimize the difference between the measured risk in a specific state of the system and the acceptable risk (i.e., the fitness function). To prove the capabilities of the proposed AIS reaction methodology, several experiments are conducted demonstrating that it is able to effectively minimize the fitness function in a more than acceptable time frame even under stressed conditions.

Future works will study the possibility of employing the proposed methodology in a real use-case scenario, studying the viability of developing the AIS-powered reaction with real network traffic in a SIEM. Besides, a meta-optimization algorithm to further enhance the selection of AIS-reaction input parameters is worth investigation. Furthermore, we will analyze the feasibility of enriching the AIS reaction with offensive countermeasures, which are considered of great interest in military scenarios.

REFERENCES

[1] A. H. Celdrán, M. G. Pérez, F. J. G. Clemente, and G. M. Pérez, ''Towards the autonomous provision of self-protection capabilities in 5G networks,'' *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 12, pp. 4707–4720, Dec. 2019.

[2] P. Nespoli, D. U. Peláez, D. D. López, and F. G. Mármol, ''COSMOS: Collaborative, seamless and adaptive sentinel for the Internet of Things,'' *Sensors*, vol. 19, no. 7, p. 1492, 2019.

[3] J. V. Botello, A. P. Mesa, F. A. Rodríguez, D. Díaz-López, P. Nespoli, and F. G. Mármol, ''BlockSIEM: Protecting smart city services through a blockchain-based and distributed SIEM,'' *Sensors*, vol. 20, no. 16, p. 4636, 2020.

[4] S. Y. Enoch, Z. Huang, C. Y. Moon, D. Lee, M. K. Ahn, and D. S. Kim, ''HARMer: Cyber-attacks automation and evaluation,'' *IEEE Access*, vol. 8, pp. 129397–129414, 2020.

[5] D. D. López, M. B. Uribe, C. S. Cely, A. V. Torres, N. M. Guataquira, S. M. Castro, P. Nespoli, and F. G. Mármol, ''Shielding IoT against cyber-attacks: An event-based approach using SIEM,'' *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–19, Oct. 2018.

[6] A. Sfakianakis, C. Douligeris, L. Marinos, M. Lourenço, and O. Raghimi, ''ENISA threat landscape report 2020,'' ENISA, Heraklion, Greece, Tech. Rep., 2020. [Online]. Available: https://www.enisa.europa.eu/publications/year-in-review

[7] J. M. Vidal and M. S. Monge, ''Denial of sustainability on military tactical clouds,'' in *Proc. 15th Int. Conf. Availability, Rel. Secur. (ARES)*, Dublin, Ireland, Aug. 2020, pp. 1–9.

[8] J. Pastor-Galindo, P. Nespoli, F. G. Mármol, and G. M. Pérez, ''The not yet exploited goldmine of OSINT: Opportunities, open challenges and future trends,'' *IEEE Access*, vol. 8, pp. 10282–10304, 2020.

[9] L. A. Vitkova, A. P. Pronichev, E. V. Doynikova, and I. B. Saenko, ''Selection of countermeasures against propagation of harmful information via Internet,'' *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1032, Jan. 2021, Art. no. 012017.

[10] P. Nespoli, D. Papamartzivanos, F. G. Marmol, and G. Kambourakis, ''Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1361–1396, 2nd Quart., 2018.

[11] A. Chowdhary, S. Sengupta, A. Alshamrani, D. Huang, and A. Sabur, ''Adaptive MTD security using Markov game modeling,'' in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2019, pp. 577–581.

[12] N. Yanes, A. M. Mostafa, N. Alshammari, and S. A. Alanazi, ''An immunity-based error containment algorithm for database intrusion response systems,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 10, pp. 167–178, 2019.

[13] B. Yang and M. Yang, ''Data-driven network layer security detection model and simulation for the Internet of Things based on an artificial immune system,'' *Neural Comput. Appl.*, vol. 33, no. 2, pp. 655–666, Jun. 2020.

[14] W. Zhou and Y. Liang, ''An artificial sequential immune responses model for anomaly detection,'' in *Proc. Genetic Evol. Comput. Conf. Companion*. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 95–96.

[15] D. A. Fernandes, M. M. Freire, P. A. Fazendeiro, and P. R. Incio, ''Applications of artificial immune systems to computer security,'' *J. Inf. Secur. Appl.*, vol. 35, pp. 138–159, Aug. 2017.

[16] P. Nespoli, F. G. Mármol, and J. M. Vidal, ''Battling against cyberattacks: Towards pre-standardization of countermeasures,'' *Cluster Comput.*, vol. 24, no. 1, pp. 57–81, Mar. 2021.

[17] S. A. Zonouz, H. Khurana, W. H. Sanders, and T. M. Yardley, ''RRE: A game-theoretic intrusion response and recovery engine,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 395–406, Feb. 2014.

[18] A. Aldweesh, A. Derhab, and A. Z. Emam, ''Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,'' *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105124.

[19] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, ''A survey of network-based intrusion detection data sets,'' *Comput. Secur.*, vol. 86, pp. 147–167, Sep. 2019.

[20] G. Gonzalez-Granadillo, E. Doynikova, J. Garcia-Alfaro, I. Kotenko, and A. Fedorchenko, ''Stateful RORI-based countermeasure selection using hypergraphs,'' *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102541.

[21] F. Li, Y. Li, S. Leng, Y. Guo, K. Geng, Z. Wang, and L. Fang, ''Dynamic countermeasures selection for multi-path attacks,'' *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101927.

[22] S. Iannucci, V. Cardellini, O. D. Barba, and I. Banicescu, ''A hybrid model-free approach for the near-optimal intrusion response control of non-stationary systems,'' *Future Gener. Comput. Syst.*, vol. 109, pp. 111–124, Aug. 2020.

[23] Y. Guo, H. Zhang, Z. Li, F. Li, L. Fang, L. Yin, and J. Cao, ''Decision-making for intrusion response: Which, where, in what order, and how long?'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[24] I. Kotenko, E. Doynikova, A. Chechulin, and A. Fedorchenko, ''Ai- and metrics-based vulnerability-centric cyber security assessment and countermeasure selection,'' in *Guide to Vulnerability Analysis for Computer Networks and Systems: An Artificial Intelligence Approach*, S. Parkinson, A. Crampton, and R. Hill, Eds. Cham, Switzerland: Springer, 2018, pp. 101–130.

[25] B. Xu, Z. Zhong, and G. He, ''A minimum defense cost calculation method for attack defense trees,'' *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Aug. 2020.

[26] B. Fila and W. Widel, ''Exploiting attack–defense trees to find an optimal set of countermeasures,'' in *Proc. IEEE 33rd Comput. Secur. Found. Symp. (CSF)*, Jun. 2020, pp. 395–410.

[27] C. Zarges, *Theoretical Foundations of Immune-Inspired Randomized Search Heuristics for Optimization*. Cham, Switzerland: Springer, 2020, pp. 443–474.

[28] W. Said and A. M. Mostafa, ''Towards a hybrid immune algorithm based on danger theory for database security,'' *IEEE Access*, vol. 8, pp. 145332–145362, 2020.

[29] C. S. K. Leung and H. Y. K. Lau, ''A hybrid multi-objective AIS-based algorithm applied to simulation-based optimization of material handling system,'' *Appl. Soft Comput.*, vol. 71, pp. 553–567, Oct. 2018.

[30] J. Chen, J. Chen, and D. Yang, ''An efficient classification algorithm based on T-Cells maturation with no parameters,'' *Int. J. Comput. Intell. Appl.*, vol. 16, no. 4, Dec. 2017, Art. no. 1750024.

[31] G. Magna, P. Casti, S. V. Jayaraman, M. Salmeri, A. Mencattini, E. Martinelli, and C. D. Natale, ''Identification of mammography anomalies for breast cancer detection by an ensemble of classification models based on artificial immune system,'' *Knowl.-Based Syst.*, vol. 101, pp. 60–70, Jun. 2016.

[32] J. Zhang, D. Li, and B. Zhao, ''A prefix hijacking detection model based on the immune network theory,'' *IEEE Access*, vol. 7, pp. 132384–132394, 2019.

[33] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, ''DeepDCA: Novel network-based detection of IoT attacks using artificial immune system,'' *Appl. Sci.*, vol. 10, no. 6, p. 1909, Mar. 2020.

[34] G. F. Scaranti, L. F. Carvalho, S. Barbon, and M. L. Proenca, ''Artificial immune systems and fuzzy logic to detect flooding attacks in software-defined networks,'' *IEEE Access*, vol. 8, pp. 100172–100184, 2020.

[35] A. Gómez-Mompeán and R. Lahoz-Beltra, ''An evolutionary computing model for the study of within-host evolution,'' *Computation*, vol. 8, no. 1, p. 5, Jan. 2020.

[36] R. Pump, V. Ahlers, and A. Koschel, ''Evaluating artificial immune system algorithms for intrusion detection,'' in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS)*, Jul. 2020, pp. 92–97.

[37] A. Chmielewski, ''Application of rough sets to negative selection algorithms,'' in *Future Data and Security Engineering*, T. K. Dang, R. Wagner, J. Küng, N. Thoai, M. Takizawa, and E. J. Neuhold, Eds. Cham, Switzerland: Springer, 2017, pp. 381–394.

[38] J. M. Vidal, A. L. S. Orozco, and L. J. G. Villalba, ''Adaptive artificial immune networks for mitigating DoS flooding attacks,'' *Swarm Evol. Comput.*, vol. 38, pp. 94–108, Feb. 2018.

[39] L. N. de Castro and F. J. Von Zuben, ''Learning and optimization using the clonal selection principle,'' *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 239–251, Jun. 2002.

[40] S. Lysenko, K. Bobrovnikova, and O. Savenko, ''A botnet detection approach based on the clonal selection algorithm,'' in *Proc. IEEE 9th Int. Conf. Dependable Syst., Services Technol. (DESSERT)*, May 2018, pp. 424–428.

[41] S. Aldhaheri, D. Alghazzawi, L. Cheng, A. Barnawi, and B. A. Alzahrani, ''Artificial immune systems approaches to secure the Internet of Things: A systematic review of the literature and recommendations for future research,'' *J. Netw. Comput. Appl.*, vol. 157, May 2020, Art. no. 102537.

[42] S. G. Bhol, J. R. Mohanty, and P. K. Pattnaik, ''Cyber security metrics evaluation using multi-criteria decision-making approach,'' in *Smart Intelligent Computing and Applications*, S. C. Satapathy, V. Bhateja, J. R. Mohanty, and S. K. Udgata, Eds. Singapore: Springer, 2020, pp. 665–675.

[43] A. Shameli-Sendi, H. Louafi, W. He, and M. Cheriet, "Dynamic optimal countermeasure selection for intrusion response system," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 755–770, Sep. 2018.

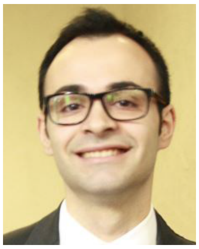
[44] J. Navarro, A. Deruyver, and P. Parrend, "A systematic survey on multi-step attack detection," *Comput. Secur.*, vol. 76, pp. 214–249, Jul. 2018.

[45] H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu, "A vulnerability assessment method in industrial Internet of Things based on attack graph and maximum flow," *IEEE Access*, vol. 6, pp. 8599–8609, 2018.

[46] A. Shameli-Sendi, M. Dagenais, and L. Wang, "Realtime intrusion risk assessment model based on attack and service dependency graphs," *Comput. Commun.*, vol. 116, pp. 253–272, Jan. 2018.

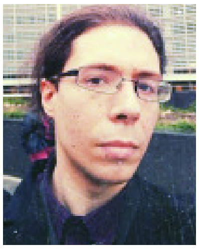
[47] H. Debar, D. A. Curry, and B. S. Feinstein, *IDMEF, Intrusion Detection Message Exchange Format*, document RFC 4765, Internet Requests for Comments, RFC Editor, 2007. [Online]. Available: https://tools.ietf.org/html/rfc4765

[48] R. Danyliw, *IODEF, The Incident Object Description Exchange Format V2*, document RFC 7970, Internet Requests for Comments, RFC Editor, 2016. [Online]. Available: https://tools.ietf.org/html/rfc7970



PANTALEONE NESPOLI received the B.S. and M.S. degrees in computer engineering from the University of Napoli Federico II, Italy. He is currently pursuing the Ph.D. degree with the University of Murcia, Spain. His research interests include ICT security, and more specifically network security, intrusion detection and response systems, and security information and event management.

FÉLIX GÓMEZ MÁRMOL received the M.Sc. and Ph.D. degrees in computer engineering from the University of Murcia, Spain. He is currently a Researcher with the Department of Information and Communications Engineering, University of Murcia. His research interests include cybersecurity, the Internet of Things, machine learning, and bio-inspired algorithms.



JORGE MAESTRE VIDAL (Member, IEEE) is currently pursuing the Ph.D. degree in computer science. He is currently a Senior Specialist in cyber defense at Indra, being part of its Digital Labs Division. He is also the Technical Coordinator of Indra's solutions for cyber situational awareness acquisition, leading the related technical activities conducted on national/international innovation programmes, like the EDA Projects Cyber Defence Situation Awareness Package—Rapid Research Prototype (CySAP-RRP) (EDA 16.CAT.OP.078) or Generation of Data Sets for Validation of Cyber Defence Tools (Cat. B FC B-1508-GP). He recently participated in the EU projects SELFNET (H2020-ICT-2014-2/671672), RAMSES (H2020-FCT-04-2015/700326), and the Full Spectrum Situational Awareness (T-SHARK) Programme of SPARTA (H2020-FCT-2015/83089).