Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Automating and Orchestrating Cybersecurity for Cyber-Physical Systems Taxonomy, Current Approaches, and Future Directions

Jamil Ahmad Kassem

jahmadkassem@uoc.edu

Universitat Oberta de Catalunya

Helena Rifa Pous

Universitat Oberta de Catalunya

Joaquin Garcia-Alfaro

Télécom SudParis

Research Article

Keywords:

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10661931/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Noname manuscript No.
(will be inserted by the editor)

Automating and Orchestrating Cybersecurity for Cyber-Physical Systems

Taxonomy, Current Approaches, and Future Directions

Jamil Ahmad Kassem · Helena Rifà Pous · Joaquin Garcia-Alfaro

the date of receipt and acceptance should be inserted later

Abstract The complex and demanding nature of security automation for CPS presents significant challenges. This is particularly true when faced with frequent and advanced attacks on numerous nodes. Performing a thorough selection of optimal defense measures to aid decision-making becomes extremely difficult to manage. This paper addresses these challenges by surveying solutions for the automation of cyber-physical security. We differentiate between two main approaches: end-to-end and modular. We also introduce a comprehensive three-phase framework to outline the main representative stages for automation of cyber-physical security. Then, we classify modular solutions according to the phase that best describes their primary contributions. Finally, we conduct an analysis that considers the merits, limitations, and outcomes of each contribution. Our study reveals opportunities for combining different modular solutions. We highlight this opportunity by forming a scoring metric for the compatibility of different modular automation approaches for combination. We provide some examples of how to integrate these solutions within our proposed three-phase framework and discuss future directions for research.

Keywords Computer security · Internet security · Cybersecurity · Cyber-Physical Security · Decision Making · Remediation

Jamil Ahmad Kassem
Universitat Oberta de Catalunya (UOC), UOC-Tech Research Center, Barcelona, Spain
CYBERCAT - Center for Cybersecurity Research of Catalonia, Spain
E-mail: jahmadkassem@uoc.edu

Helena Rifà Pous
Universitat Oberta de Catalunya (UOC), UOC-Tech Research Center, Barcelona, Spain
CYBERCAT - Center for Cybersecurity Research of Catalonia, Spain
E-mail: hrifa@uoc.edu

Joaquin Garcia-Alfaro
SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France
E-mail: joaquin.garcia_alfaro@telecom-sudparis.eu

1 Introduction

In the specific case of operational problems involving computer and Internet security stakeholders have traditionally focused on implementing predefined defense measures to address known attack vectors and threat signatures [1]. Defense measures are a set of actions that are implemented on the system to mitigate or recover from an attack. While practical, such technologies rely heavily on the assumption that defenders are aware of adversarial tactics in advance. This assumption is unrealistic given the dynamic and evolving threat landscape [2, 3]. As such, automation becomes crucial in cybersecurity scenarios, specifically for the selection of defense measures in Cyber-Physical System (CPS). Especially when CPS networks were designed primarily with a focus on performance rather than security [4]. These factors underscore the need for automated systems that can effectively manage security and keep pace with evolving threats. This problem is more relevant when considering the large number of expected devices in the near future [5]. This widespread adoption highlights the growing presence and impact of CPS technologies across different industries and sectors.

In addition to the widespread use of CPS devices, the sensitive data they hold makes them an attractive target for attacks and malicious activities. Beyond the physical devices, attacks may involve extracting sensitive information, extorting a ransom, or causing direct harm [6]. Several strategies have already been introduced in the scientific literature for the mitigation of these attacks. However, all these strategies suffer from significant drawbacks with a lack of systems capable of handling the various network configurations without expert intervention.

Automation can intervene at multiple stages of the cybersecurity life cycle. It must ensure the deployment of robust and concrete measures across representative stages of potential cyber incidents. Examples of these measures include: prevention and protection mechanisms (e.g., cryptography and access control [7, 8]); monitoring and mitigation mechanisms (e.g., pattern matching and reconfiguration theories [9, 10]); and post-attack analysis and recovery mechanisms (e.g., forensics and adaptation techniques [11, 12]). Previous surveys [13] studied automation solutions separately and as holistic systems. In this survey, we try to form a basis for modulating automation. Modularity is built on the basis of scoring solutions and finding an index of cooperation. Building upon previous knowledge from those stages mentioned above, and aiming at revealing new elements and future research directions, the primary contributions of this paper are outlined as follows:

1. **Categorization of cybersecurity automation**. We categorize security automation systems into three main operational phases. This categorization clarifies how automation challenges are addressed. We also provide a foundational understanding of system performance metrics.
2. **Analysis of current drawbacks and automation challenges**. The strengths and weaknesses of existing automation approaches are examined

and cross-referenced with each other. This analysis highlights a common challenge across all methods and offers a comprehensive view of the limitations of current automation frameworks.

3. **Development of a compatibility framework for modular automation.** To address the challenges of integrating multiple automation approaches, we introduce a novel compatibility scoring framework. This framework formally evaluates modular automation approaches against eight key metrics, including *code availability*, *I/O compatibility*, and model *customizability*. The resulting score quantifies the ease with which different systems can be combined. The metric provides a clear methodology for building effective security systems from modular components.
4. **Identification of future research directions**. Building on identified challenges and aligning with cybersecurity objectives, potential research avenues are outlined. These include areas that require further investigation to overcome current limitations and improve the efficacy of automation systems in cybersecurity contexts.

The remainder of the paper is organized as follows. Section 2 presents the literature review methodology followed in our study. Section 3 introduces our three-phase categorization framework for automation and selection of security measures. Section 4 presents our main findings, which are further discussed in Section 5 for future research directions and open problems. Section 6 concludes the paper.

2 Literature Review Methodology

The survey starts with Section 2.1 that formulates the research questions that we want to answer. Section 2.2 illustrates the research strategy and the sources for the articles. Section 2.3 presents the method we used to search for relevant publications. Section 2.4 establishes the technique used to include and exclude criteria for the articles. Finally, Section 2.5 introduces the article screening procedure.

2.1 Research Questions

This study aims to systematically analyze the role of automation in cybersecurity, particularly in the context of CPS environments. To guide our investigation, we address the following research questions:

RQ1. What are the current approaches for automation of cybersecurity, and which primary approaches do they employ?
RQ2. What are the relationships among the identified approaches, and how can they be effectively integrated within a unified system?
RQ3. What criteria can be used to evaluate the performance and compatibility of different approaches?

RQ4. How can the identified approaches be systematically categorized in a way that supports their *combined deployment* within cybersecurity frameworks?
RQ5. What are the functional advantages and limitations associated with each approach?
RQ6. What opportunities exist for enhancing cybersecurity automation, and which directions should future research explore?

2.2 Search and Selection Strategy

We conducted a survey following the PRISMA 2020 guidance for identification, screening, and selection of studies. Our protocol was designed to answer RQ1–RQ6 and was applied consistently across three databases. We searched three major digital libraries:

- ACM Digital Library: A key resource for computing and information technology research, featuring journals, conference proceedings, and technical magazines from the Association for Computing Machinery [14].
- IEEE Xplore: A wide repository for publications in electrical engineering, computer science, and related fields, including journals, conference proceedings, and standards, with comprehensive abstract and full-text search [15].
- Scopus: One of the largest abstract and citation databases, covering a broad range of scientific, technical, medical, and social sciences literature [16].

For each database we exported the results of the search as RIS files, which we then post-processed programmatically. In total this yielded 109 IEEE records, 28 ACM records, and 243 Scopus records, for a combined 380 records prior to de-duplication. RIS files were parsed and individual records were identified. For each record we extracted the DOI and title. We defined a canonical key per record as the DOI (lower-cased) when available, and otherwise the normalized title. Records sharing the same key were considered duplicates. Using this rule, we removed 63 duplicate records and retained one canonical record per DOI/title, resulting in 317 unique records for screening.

2.3 Search Query

To collect relevant and recent literature, we consulted the three aforementioned databases with a precise query to capture publications related to cybersecurity, automation, and CPS during the last 15 years. Table 1 outlines the structure and main components of the precise query.

2.4 Inclusion and exclusion criteria

We then applied explicit inclusion and exclusion criteria at title/abstract level. A study was considered in scope if:

Table 1: Query values for our systematic literature review

Concept	Field	Value(s)
Cybersecurity	Title	`"computer security" OR "Internet security" OR cybersecurity OR "cyber security" OR "information security" OR "network security" OR infosec`
Automation	Abstract	`automation OR "automatic response" OR "autonomous system*" OR "adaptive security" OR orchestration OR management OR "self-healing" OR "self-defending"`
CPS	Title	`"cyber-physical system*" OR cps OR "smart device*" OR "edge device*" OR iot OR "internet of things"`
Years	Year	`PUBYEAR > 2010 AND PUBYEAR < 2026`
Language	Language	`English`

– It targeted cyber-physical systems, industrial control systems, SCADA, smart grids, or closely related CPS domains.
– It proposed, evaluated, or analysed security automation mechanisms in at least one of our three phases (discovery/detection, scoring/risk assessment, selection and planning response orchestration). Examples include automated response selection, autonomous intrusion response, security orchestration and playbooks, or runtime enforcement.

We excluded studies at this stage if any of the following conditions applied:

1. The work did not address security automation, orchestration, or response (for example, pure anomaly detection, vulnerability analysis, or threat modeling without automated defense selection) (removed 51).
2. The work did not clearly cover CPS, ICS, SCADA, smart grid or other industrial CPS contexts (generic IT, cloud, and web security without a CP component) (removed 60).
3. The paper was a survey or review (e.g., "Security Modelling for Cyber-Physical Systems: A Systematic Literature Review"). Relevant Surveys are discussed in related work but not coded into the main work (removed 36).
4. The paper introduces a testbed, testing environment, or a dataset that can be used for cybersecurity (removed 53).
5. The work addressed only a single attack type with non-generalizable metrics (e.g., mitigation strategies that rely solely on the number of requests for a specific DoS variant or a tightly tailored false-data-injection scenario) and did not present a mechanism that can be extended to other attacks (removed 72).

2.5 Automated title/abstract screening

To make the screening reproducible, we implemented the inclusion/exclusion rules as a simple classifier over titles and abstracts. For each unique record we

constructed a lower-cased string combining the title and abstract and applied the following filters sequentially:

1. Security automation / orchestration: records were required to contain at least one automation-related term ("automation", "orchestration", "runtime enforcement", "autonomous response", "security orchestration", "SOAR", "automated mitigation", "intrusion response", "playbook"). Studies that only described detection, modelling, or static hardening were excluded at this stage.
2. CPS/ICS context: records were required to mention at least one CPS-specific term ("cyber-physical", "industrial control", "ICS", "SCADA", "smart grid", "power system", "PLC"). Studies without such terms in the title or abstract were excluded as out of scope.
3. Survey/review detection: records containing phrases such as "survey", "systematic literature review", or "a review of" were excluded as surveys.
4. Attack-specific non-generalizable mitigation: records mentioning only one specific attack type with tailored metrics (e.g., "denial of service", "DDoS", "false data injection", "command injection", "response injection") and no indication of a general response framework were excluded as attack-specific and non-generalizable.

The remaining checks were done manually by going through titles and then fine tuning through abstracts. This procedure corresponds to PRISMA's screening and eligibility stages at the title/abstract level.

The 26 remaining papers constitute the automation corpus used in our qualitative synthesis and categorization. For each of these studies we extracted data and formulated a study found in the conclusions and recommendations of this paper.

3 Security Automation and Defense measures

Building on the conducted literature review, CPS solutions are fundamentally divided into end-to-end and modular automation. Effective security automation relies on several interconnected subsystems serving as the fundamental building blocks for mitigating vulnerabilities and attacks. These blocks include passive elements (e.g., databases and logs) and active elements (e.g., monitoring and defense enforcement blocks). The functions of these building blocks are categorized into three distinct phases that define the types of automation. End-to-end automation includes all three phases and can achieve full automation alone. Modular automation covers two or less phases. The three phases are explained in the following.

1. **Discovery and Detection**: It includes the deployment of approaches to monitor for vulnerabilities and identify potential adversarial activities. Ultimately, it seeks to approximate the attack methodologies currently employed or intended by an adversary.

2. **Scoring**: It is used to quantify associated costs with potential defense measures, f.i., by taking into account a set of policies and system performance metrics (e.g., to estimate the cost of enforcing such defense measures into the system). The score can also indicate the effectiveness of those defense measures, or to anticipate potential collateral damage. These scores provide some precise decision elements to the eventual stakeholders who must enforce the best defense strategy.
3. **Selection and Planning**: It involves algorithms and heuristics to shortlist and propose the most appropriate plan to remediate the security issues previously reported. The rationale is to maximize security, regardless of the attack status. The goal could either be to stop the attack completely or to reduce its impact upon the system, while searching for a better solution.

Table 2: Summary of selected publications, ordered chronologically and alphabetically. We start with End-to-End followed by modular publications

Publication	Database
01. Poolsappasit *et al.* [17] (2012)	IEEE Xplore
02. Kordy *et al.* [18] (2014)	IEEE Xplore
03. Nespoli *et al.* [19] (2021)	IEEE Xplore
04. Stan *et al.* [20] (2022)	IEEE Xplore
05. Alevizos *et al.* [21] (2025)	Scopus
01. Motzek *et al.* [22] (2017)	Scopus
02. Martinelli *et al.* [23] (2018)	Scopus
03. Shameli *et al.* [24] (2018)	IEEE Xplore
04. Soikkeli *et al.* [25] (2019)	ACM Digital Library
05. Yaqoob *et al.* [26] (2019)	Scopus
06. Gonzalez *et al.* [27] (2020)	Scopu
07. Li *et al.* [28] (2020)	Scopus
08. Haque *et al.* [29] (2021)	Scopusk
09. Kavallieratos *et al.* [30] (2021)	Scopus
10. Paul *et al.* [31] (2021)	Scopus
11. Stan *et al.* [32] (2021)	IEEE Xplore
12. Uuganbayar *et al.* [33] (2021)	Scopus
13. Sawik [34] (2022)	Scopus
14. Ameri *et al.* [35] (2023)	ACM Digital Library
15. Bringhenti *et al.* [36] (2023)	IEEE Xplore
16. Kilincer *et al.* [37] (2023)	Scopus
17. Settanni *et al.* [38] (2023)	IEEE Xplore
18. Yang *et al.* [39] (2023)	ACM Digital Library
19. Basak *et al.* [40] (2024)	Scopus
20. Phillips *et al.* [41] (2024)	IEEE Xplore
21. Coscia *et al.* [42] (2025)	Scopus

The three phases can be directly linked to the automation framework and the defense measures. These dual categorizations allow the establishment of a direct relation between automation and defense measures. In addition, automation is also classified by its level of autonomy, which is defined by the required degree of human interaction and ranges from minimal to total automation. These categorizations have already been covered in different surveys [13]. In

our survey, we present this classification as end-to-end and modular to bridge the gap between the two classes as presented in Section 4.2.

Based on the methodology defined in Section 2, we conduct a comprehensive review of the literature to identify contributions to the field of automation. A quick summary of the selected publications is shown in Table 2. We examine these publications in more detail and identify their main contributions. Of these publications, we separate those that handle end-to-end automation and present them in Section 3.1. The remaining publications are methodically categorized in Section 3.2 according to the phase that most accurately characterizes their main contribution. For each article, a brief explanation of the system functionality is provided. Finally, we develop a discussion in Section 4 where we present our findings, identify existing shortcomings, and propose potential areas for future research in Section 5.

3.1 End-to-end Automation Approaches

We first examine the selected publications that match the concept of end-to-end automation (i.e., including all phases of an automation process). We selected them as representative systems engineered to integrate seamlessly with an organization's existing security framework. These approaches leverage data from various sources to automate cyber management operations. We group them into three main trends: Logic and probabilistic-based decision frameworks, biology-inspired defense approaches, and decentralization approaches.

We start by reporting work on logic-based models to construct support decision tools that allow security analysts to reason systematically about possible attack paths, vulnerabilities, and defense strategies. Kordy *et al.* [18] work on attack-defense trees, as an extension of earlier attack tree models by Schneier [43]. The work covers both attacks and their corresponding mitigation strategies. Defense measures can be placed anywhere in the tree, allowing for a more accurate modeling of the adversary and the defender. Goals are represented by root nodes while branches represent attacks and defense actions. Follow-up work (e.g., Xu *et al.* [44]) utilizes algebraic algorithms to compute costs and transform trees into graphs, ending by the identification of optimal defense paths.

Stan *et al.* [20, 32] extend advanced work on attack graphs, enhancing and improving MulVAL [45], a well-known logic-based attack graph engine for expert systems. MulVAL represents adversarial activities as logical preconditions for attack steps. The new framework from Stan *et al.* focuses on network architectures and protocol vulnerabilities associated with advanced network architectures. It includes an understanding of layered network strategies, coupled with a comprehensive modeling of network attack scenarios. Specifically, the new framework considers spoofing, Denial-of-Service (DoS), Man-in-the-Middle (MitM), and wireless or bus communication attacks. In a different vein, Poolsappasit *et al.* [17] propose the use of probabilistic-based attack engines, extending the use of Bayesian attack graphs for security assessment.

Their probabilistic model offers a detailed understanding of system features to provide dynamic evaluation and future prediction of attack steps. The work relies on an in-house risk analysis tool using the NIST Common Vulnerability Scoring System (CVSS) framework. Combined with a vulnerability scanner, network topology traces, and a sequence of vulnerability database queries, the system conducts a multi-step risk assessment process for attack mitigation and automation. The system is defined through a probability-based index calculation that encompasses various aspects, including the risk and cost of attacks and defense actions. During optimization a pool of candidate solutions is offered within an iterative selection process.

Moving away from trees- and graph-based tools, Nespoli *et al.* [19] propose an automation approach that artificially mimics a biological immune system. In their immune system proposal, defense measures act like antibodies and are selected using a set of indices (e.g., cost of implementing, effectiveness, and impact of the defense measure). The variables are combined to give a numerical index of the effectiveness [46]. The defense measures may operate in static mode where the system threat set is predefined. Another operation mode is the dynamic mode where threat sets can vary. The algorithm also considers the cross-relation between defense measure approximated to the average of none and perfectly combinable. Where a perfectly combinable index is calculated by the sum of the two defense measures indices, capped by a maximum value. The non-combinable index is simply the max of the two defense measures indices.

The algorithm is capable of analyzing the system and identifying vulnerabilities and attacks for each resource. These attacks are associated with a set of antibodies selected based on a threat level metric. The threat level is a combination of the value of the targeted resource, the probability of occurrence, the impact, and the effectiveness of the antibodies. The likelihood of threat occurrence and the impact of threat are derived from the CVSS. By comparing the threat level to the value of each resource, the authors establish a benchmark for the effectiveness of implementing the selected antibody set.

A final automation approach moves away from the vulnerability-oriented defense to a more organization policy-oriented study. Alevizos [21] develops a framework for automating security via blockchain technologies. Smart contracts are proposed for the automation of both the enforcement and verification of security policies. The use of Machine Learning (ML) is suggested to identify anomalies that are subsequently used for updating or adjusting defense measuresthat are documented on a distributed ledger. Automated scripts are employed to scan network traffic, user behavior, and both external and internal threat intelligence. The overall approach combines the NIST Cybersecurity Framework [47] and ISO 27001 guidelines [48] to create policies tailored for a small-sized organization, using the MITRE ATT&CK framework [49] as a knowledge base. An event-based trigger mechanism responds to security events as they occur. Remediation is handled by an enforcement engine that ingests NIST guidance for simplicity.

3.2 Modular Automation Approaches

In contrast to end-to-end approaches reported in the previous section, most automation solutions reported in the literature focus on specific but isolated problems. They provide modular approaches designed to be highly effective at their particular tasks and can often be integrated into larger security ecosystems. This integration allows organizations to adopt automation in a more gradual and targeted manner, addressing their most pressing security challenges first. Some examples of the kind of modular automation approaches we refer to are listed next:

- Discovery of Vulnerabilities: Automated scanners continuously probe networks and applications to handle existing vulnerabilities.
- Threat Analysis and Attack Detection: Specialized tools use automation to detect known patterns or to report anomalies that can be associated with threats and attacks.
- Identity and Access Management: Both can be associated with formal policies to identify deviation from legitimate actions, as well as reporting strategies to mitigate attacks and enforce remediation plans.
- Compliance and Auditing: Automated tools shall continuously monitor systems and configurations for compliance with policies and best practices, before, during, and after the perpetration of adversarial activities.

The following sections iterate through the remaining publications listed in Table 2 for modular automation. In turn, these publications are arranged next by one of the three stages of automation, for which these publications on modular automation approach seem to apply.

3.2.1 Discovery and Detection

During the discovery phase, the main objective of the defender is to identify adversaries and their intents. Detection is the most research-rich phase with automation components ranging from traditional methods for pattern matching to recent trends for anomaly detection. There is vast literature on such topics from the construction of legacy Intrusion Detection System (IDS) to more advanced solutions, such as ML pattern recognition. We explore these components in the realm of cybersecurity automation.

Kilincer *et al.* [37] study the use of multiple datasets and ML techniques for detection in the Internet of Medical Things. The authors explore two recursive feature elimination methods: logistic regression and extreme gradient boosting regression. They evaluate their approach across four diverse datasets: the Intensive Care Unit, TON-IoT, WUSTL-EHMS healthcare monitoring, and pre-optimized ECU-IoHT datasets. Feature optimization is completed using k-nearest neighbor and a hyperparameter-tuned multilayer perceptron classifier via Grid-Search CV. Automation is further extended to network management by Yang and Shami [39], who propose an autonomous AutoML-based IDS

framework for 5G/6G environments. The pipeline automates data preprocessing, feature engineering, base model learning, hyperparameter optimization, and model ensembling. Class imbalances are handled through a Tabular Variational Auto-Encoder, while automated feature selection narrows down the best contributors from tree-based models before performing a confidence-based stacking ensemble.

For end-host defense, Basak *et al.* [40] present CyberSentinel, a transparent framework for image-based malware detection. Raw malware binaries are mapped into uniform 2D blocks to optimize structural feature extraction. These 2D blocks are treated and can be visualized as images using RGB color mapping. The architecture employs a two-branch deep network integrating an asymmetric spatial attention mechanism alongside a reduced-resolution auxiliary branch to accelerate performance and prevent overfitting.

The integration of network security enforcement and threat sharing is another strong detection approach. Haque *et al.* [29] propose an automated defense system leveraging STIX and TAXII standards to convey structured Cyber Threat Intelligence. Threats are automatically detected and managed using a REST Application Programming Interface (API) paired with a Relationship-Based Access Control model to govern safe information exchange. To handle active threats, they introduce automated course-of-action tasks ranging from automated mitigations to employee updates.

For policy verification, Bringhenti *et al.* [36] showcase VEREFOO, a framework for the automated orchestration of virtual distributed firewalls. Taking network topologies and abstract security intents as inputs, the system models the orchestration as a constraint programming problem. Solving this optimization problem guarantees the formal correctness of the generated rule sets while minimizing the firewall allocation footprint. Finally, reputation metrics are used by Coscia *et al.* [42] with APIARY, an automated YARA rule generator. By identifying distinctive API patterns that differentiate malware from goodware. The algorithm scores APIs and builds concise signature models that optimize the trade-off between detection rates and false positives.

3.2.2 Scoring

Before an automation approach can orchestrate and execute a defensive strategy, it must evaluate defense measures to establish a prioritization index. This prioritization allows the system to map dependencies and efficiently schedule defense measures deployments. To achieve this, diverse analytical frameworks have been proposed in the literature to mathematically score, compare, and model defense measure effectiveness.

A prominent line of research focuses on quantitative financial assessment indices, including Return on Investment [50], Return on Response Investment (RORI) [51], and Return on Security Investment (ROSI) [52]. These metrics evaluate the financial feasibility of defense alternatives within complex environments. However, a major challenge lies in fitting these metrics to dynamic systems, as parameters often demand context-specific tuning. To bal-

ance technical side-effects against economic metrics, Motzek *et al.* [22] formulate a multi-dimensional optimization framework to define Pareto-efficient response plans. The model assesses two orthogonal metrics. A Response Financial Impact Assessment, which combines RORI metrics with an attack volume geometric model. A Response Operational Impact Assessment which maps critical asset and mission dependencies. By plotting these conflicting vectors, the system helps operators identify solutions that maximize risk mitigation while minimizing faults to functions. Gonzalez-Granadillo *et al.* [27] expand on this paradigm by presenting a Stateful RORI model capable of capturing temporal and sequential system changes. Using a directed hypergraph representation, their approach models structural restrictions, interdependencies, and operational conflicts of defense measures across historical system states. This hypergraph structure prevents defensive over-provisioning and accounts for active versus passive defense measures states. Transitioning to predictive asset defense, Yaqoob *et al.* [26] design a quantitative framework to optimize ROSI. The model incorporates asset values, vulnerability scan profiles, and threat modeling. This calculation leverages a Bayesian probability architecture derived from the CVSS and historical network logs, offering an investment recommendation.

Alternative automation frameworks prioritize scoring approaches driven by structural network graphs and dynamic state parameters. Shameli-Sendi *et al.* [24] introduce a defense-centric dynamic framework mapped through a combination of Attack-Defense Trees and Service Dependency Graphs. Their multi-objective optimization engine scores defense measures considering execution costs, performance costs, and quality-of-service penalties. Taking a broader time-horizon perspective, Soikkeli *et al.* [25] construct a cost-impact model designed to account for cascading lateral compromises and service availability losses. Their approach measures three core cost indicators: availability degradation, immediate action, and long-term system recovery costs. Their work demonstrates that allowing controlled attack progression through non-critical nodes while prioritizing service restoration can yield to lifetime cost efficiency. In the context of interconnected logistics, Sawik [34] models cybersecurity risk allocation within multi-tier supply chains using Stochastic Mixed-Integer Programming. The framework applies conditional maxmin and minmax optimization rules to analyze direct and indirect cyber threats. Rather than viewing system nodes uniformly, positioning within a vendor hierarchy influences an asset's attractiveness, prioritizing investing toward weak points.

Finally, recent literature explores knowledge-based, semantic scoring mechanisms that extract requirements from documentation or simulate structural dependencies. Ameri *et al.* [35] present an information system designed for the semi-automated management of industrial control system security. The framework leverages Named Entity Recognition and Natural Language Processing models to automatically extract security capability claims and regulatory compliance parameters. These unstructured text pieces are transformed into machine-readable verification profiles, allowing security teams to audit deployed configurations against documented claims. Addressing complex cyber-

physical interdependencies, Phillips *et al.* [41] present a simulation-based approach to automate risk assessments in compliance with ISO 27005 guidelines. The core engine applies systemic cause-and-effect ontology modeling to categorize assets into structured classes. Relationships are mapped among assets, trustworthiness levels, potential threats, and security controls. The simulation automatically isolates secondary effect trajectories and hidden attack paths while providing an automated risk matrix.

3.2.3 Selection and Planning

During the selection phase, the automation system determines and orchestrates the most effective defense strategy to remediate adversarial activities. The system must assess and select a wide range of defense measures within strict time windows.

Several approaches rely heavily on mathematical modeling and risk metrics for countermeasure selection. Paul and Zhang [31] present a decision-support model formulated as a two-stage stochastic programming framework to minimize total social cost. The first stage addresses proactive defense prior, informed by positive externalities and government intelligence sharing. The second stage handles post-attack reactive defense measures, allowing firms to dynamically optimize their security portfolio under uncertain threat landscapes.

To handle multi-step, multi-path attacks, Li *et al.* [28] propose a dynamic defense measure selection mechanism using a Probabilistic Attack-Response Tree. The framework balances three core parameters: deployment costs, security benefits, and quality-of-service impacts. By utilizing Naive Bayesian theory to estimate multi-path attack probabilities, their greedy selection algorithm iteratively optimizes the activated defense measures. Moving into cyber-physical domains, Kavallieratos *et al.* [30] combine structured threat modeling with multi-objective metaheuristic optimization. They utilize the STRIDE and DREAD methodologies to systematically identify vulnerabilities and quantify risk propagation across system components. A Non-dominated Sorting Genetic Algorithm II is then deployed to select defense measures, balancing defense and enforcement costs.

Other approaches frame selection around alternative graph-theoretic paradigms. Stan *et al.* [32] employ a heuristic search methodology on logic-based attack graphs generated by MulVAL. They map the network's risk profile into a state-space pathfinding problem where graph edges signify adversarial actions or vulnerability exploitations. By weighting these paths according to the CVSS and applying the A* search algorithm, the framework isolates the optimal, budget-constrained mitigation plan.

In terms of financial planning, several authors shift away from conventional approaches asset protection toward integrated risk-transfer models. Martinelli *et al.* [23] model the economic trade-offs between implementing technical security controls and purchasing cyber insurance to mitigate losses. They formulate this co-optimization problem as a multi-dimensional knapsack vari-

ant, showing how technical investments drive down insurance premium rates. Uuganbayar *et al.* [33] expand by introducing advanced greedy and genetic algorithms designed to maximize an organization's expected utility. Their approach captures non-linear premium reduction functions and evaluates defense measures based on their direct impact on asset vulnerability and insurance coverage boundaries. Finally, knowledge-based automation approaches focus on semantic selection rather than purely mathematical optimization. Settanni *et al.* [38] propose a model for automated approach that translates high-level security capabilities into abstract mitigation recipes. Written in a domain-specific, machine-readable language, these recipes are parsed, contextualized against a graph-based network topology, and automatically compiled into standardized OASIS CACAO security playbooks. This allows virtualized network security functions to dynamically deploy configuration rules.

4 Discussion

This research aims to construct a comprehensive knowledge concerning the inherent strengths and weaknesses associated with cybersecurity automation approaches. Section 4.1 starts by studying end-to-end automation systems. A compatibility based on eight metrics is introduced in Section 4.2. A categorization of the surveyed work is formed in Section 4.3. The subsequent Section 4.4 studies the surveyed work from an analytical perspective. Section 4.5 ends with the found strengths and weaknesses.

4.1 End-to-end Automation

Given the complexity of end-to-end systems, our survey identified only five contributions that cover the three phases of automation. Moreover, each system presents a unique contribution that distinguishes it from the others. This variation affords security managers bigger margins in selecting a solution tailored to specific applications.

Starting with the efforts by Kordy *et al.* [18] and Stan *et al.* [20], to improve decision support tools' compatibility and model generation. The authors augment the scalability limitations of previous approaches, thereby addressing complexity issues. Similarly, Poolsappasit *et al.* [17] enhance the usage of probabilistic attack graphs with Bayesian theories, thus offering dynamic capabilities that can complement the prediction of attack vectors. These approaches target the main weakness in the use of tools like attack graphs found in the exponential growth of complexity with network size. Subsequent systems, like the one introduced by Nespoli *et al.* [19], draw inspiration from the robustness of the biological immune system in addressing complex defense challenges. This implementation enables the system to sustain high robustness in selection defense measures by explicitly considering the correlation between distinct defense measures components. The reframing of the problem still lacks

an answer to the issue of detection where the system performance heavily depends on the detection of adversarial actions. Another issue is found in the metrics used where though the model is innovative it is not as accurate or inclusive as RORI and ROSI. In the fourth and final system by Alevizos *et al.* [21], the authors successfully bridge the gap with the practical application of cybersecurity. By accounting for organizational policy requirements, a solution can be provided that is more feasible to implement from a managerial perspective. This approach is more concerned with maintaining functionality from a micro level. As such it puts the system at risk where critical functionality is targeted or persistent attacks occur.

Table 3: Checklist used to assess the compatibility of each approach along eight criteria. Each row lists concrete conditions that can be verified from the available artefacts (e.g., code repository, documentation, or published paper).

Criterion	**Checklist items**
Code Availability (CA)	CA1: Source code is publicly accessible (repository or archive). CA2: Scripts or documented commands to reproduce parts of the approach. CA3: A licence or usage statement explicitly allowing research use is provided.
Data Source (DS)	DS1: The primary data (or a close equivalent) is publicly accessible. DS2: The data schema (fields, types, and key attributes) is documented. DS3: Any preprocessing or transformation steps are described or scripted.
Model Customizability (MC)	MC1: Behaviour can be directly configured (configuration files). MC2: The implementation exposes modular components (plugins or hooks). MC3: Thresholds, models, or rules can be replaced or adjusted.
Support (SP)	SP1: Dedicated documentation (user guide, API reference, or tutorial). SP2: Visible activity in the last 12 months (commits, releases, or responses). SP3: Active support or feedback channel (e-mail, forum, or contact address).
Automability (AU)	AU1: Executed non-interactively (CLI, API, or script). AU2: Batch processing or pipeline integration. AU3: Does not require GUI interaction for typical use.
I/O Compatibility (IO)	IO1: Inputs are accepted in standard formats (JSON, CSV or standard APIs). IO2: Outputs are produced in standard, machine-readable formats. IO3: Inputs and outputs formats are documented (schemas or descriptions).
Environment (EN)	EN1: Can be installed (package manager, container image, or install script). EN2: Does not require specialised hardware. EN3: Does not depend on a proprietary or highly constrained stack.
Latency (LT)	LT1: The authors report runtime or measurements of typical workloads. LT2: The implementation supports non-blocking or streaming processing. LT3: Reported runtimes are consistent with intended operational use.

4.2 Compatibility of Modular Approaches

As end-to-end automation contributions are increasingly complex and rare, utilizing modular automation becomes a more viable strategy. However, the effectiveness of such integration depends heavily on the selection process and how compatible different systems are. In modular automation, components may originate from multiple vendors, use varied data formats, or support different levels of decision-making autonomy. Without a framework to assess how

well these systems cooperate, their integration may result in inefficiencies, security gaps, or conflicting actions.

To quantify this integration potential, we introduce a *compatibility scoring score* c_i defined along eight criteria, which we present in Table 3. Each c_i is assessed using three binary conditions. For every approach, we determine for each c_i how many checklist items are satisfied and derive a compatibility along that dimension. This procedure makes the scoring rule deterministic: given the same documentation, any application of the checklist will arrive at the same categorical rating.

To obtain an overall compatibility value for a given approach, we first quantify each c_i as the sum of binary conditions divided by three. The overall compatibility score C is the arithmetic mean of the eight normalized criterion scores: $C = \frac{1}{8} \sum_{c=1}^{8} c_i$.

We deliberately adopt an additive model with equal weights because the c_i capture complementary aspects that are jointly necessary for practical integration. Equal weighting is a common and transparent choice in multi-criteria assessments when there is no strong a priori reason to privilege one dimension over another.

To validate the applicability and robustness of this scoring model, we applied it to the cybersecurity automation systems discussed in Section 3.2. Each system was evaluated against the eight checklist-based criteria. The resulting scores were visualised using radar charts in Figures 2, 3, and 4, which compare how well different approaches support modular integration across the considered dimensions. A detailed table outlining the justification for each score per system is available online in a companion GitHub repository.[1]

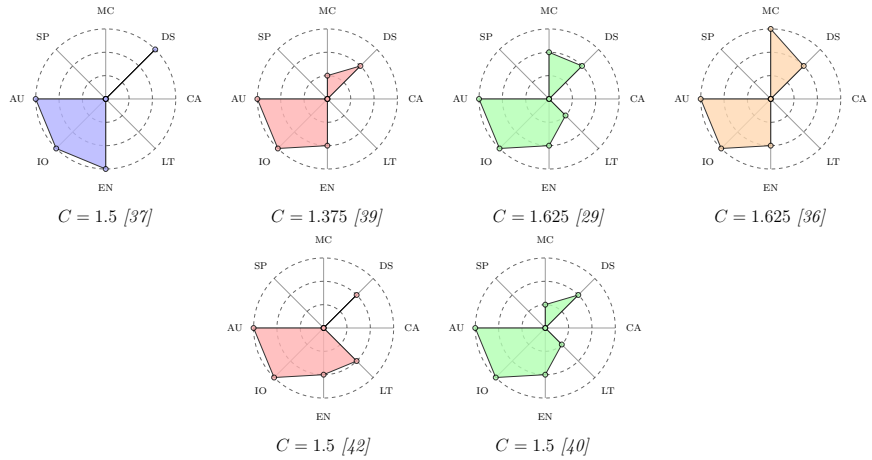


Fig. 2: Radar of the compatibility of the detection automation approaches.

[1] Available online at: `https://github.com/JamilahKassem/compatibility-cybersecurity-automation-survey`

- **Discovery and Detection.** Systems scored high in *Automability* and *I/O*. We attribute this to the main role of this phase, which is to prepare for mitigation and response. This phase is also good at *Data* and *Environment* showing that on average this phase adapts well to different working conditions. With an overall average score of 1.52, these approaches are flexible enough for many uses with only minor adjustments. Their design focuses on adaptability, quick integration, and minimal disruption with an average at 1.52.
- **Scoring.** The highest score in this phase is for *Environment*, at 2.125. Researchers clearly emphasized user flexibility when designing these solutions. Most approaches in this category offer several configuration options. Additionally, *Data Source* was also high at 1.875 due to the importance of data when scoring defense measures. Another strong score is found in *Automability* where researchers emphasized the independence of scoring with a small dependence on the detection techniques. However, less attention was given to integration with low scores in *I/O* and *Complexity*. This especially true with approaches that prioritized accuracy with the consideration of multiple metrics while disregarding the standardization and availability of these metrics. As a result, many of these solutions are complex to implement. The systems do not use popular tools and are slow to deploy, as shown in the low *Latency* and *Customizability* scores. Overall, the scoring-phase approaches show low compatibility with an average at 0.98.
- **Selection and Planning.** The results in this phase shows best scores in *Data Source* and *Environment*. Similar to scoring, selection is highly dependent on data making it important. Secondly, researchers design these approaches irrelevant of network making them highly applicable thus the high *Environment* score. This phase scored low in *Customizability* and *I/O* as researchers treat this phase as a separate end of automation phase with little support to subsequent actions. These results suggest that while the solutions are functional, they are less suited for modular setups. The overall phase shows the lowest compatibility with an average score of 0.82.

For the remaining four metrics, we form conclusions that are shared across all the publications.

- **Code.** Out of the 21 reviewed modular automation publications, only four provide openly available source code (average 1.1). The rest either do not mention code availability or only share it partially.
- **Data.** In most cases (average 3.2), the datasets used are either publicly accessible or easy to reproduce. This availability supports the reusability and adaptability of the proposed solutions.
- **Support.** A common issue across all publications is the lack of long-term support (average 1.3). This is primarily due to the academic nature of the research, which often relies on short-term or part-time funding. Without backing from large enterprises or long-term projects, sustained support remains unlikely.

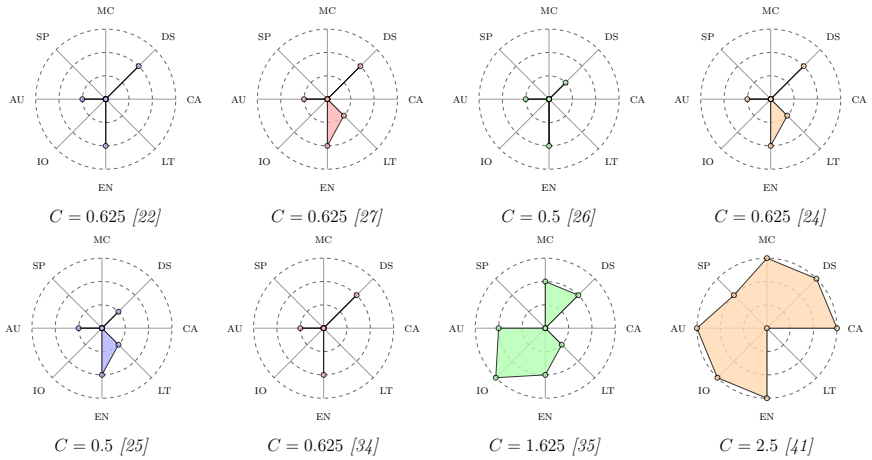


Fig. 3: Radar of the compatibility of the scoring automation approaches.

– **I/O.** Most approaches demonstrate strong flexibility in handling input and output data (average 3). This flexibility makes them easier to integrate into larger, more complex systems.

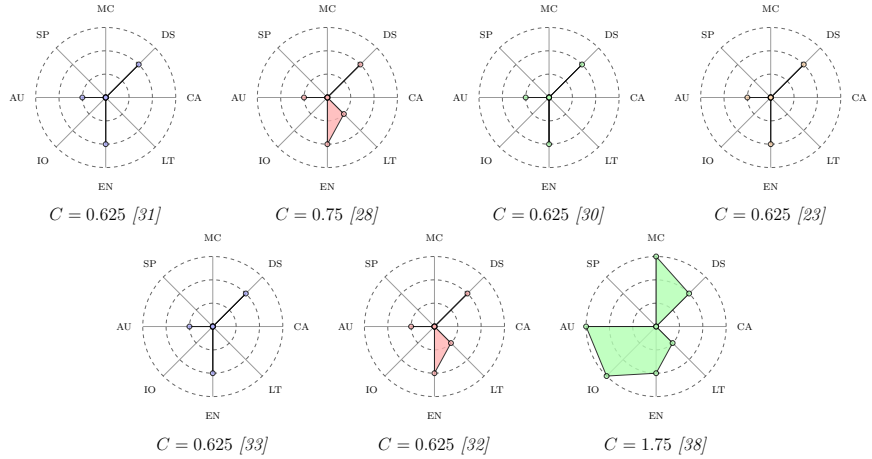


Fig. 4: Radar of the compatibility of the selection automation approaches.

4.3 Modulating Automation

To facilitate the understanding of modular automation, we form the diagram in Figure 5. The diagram illustrates our phase-based framework for automat-

ing cybersecurity. It starts with the input available before automation and ends with applying the defense actions and defense measures. Each phase is supported by a set of techniques, which are shown alongside it.

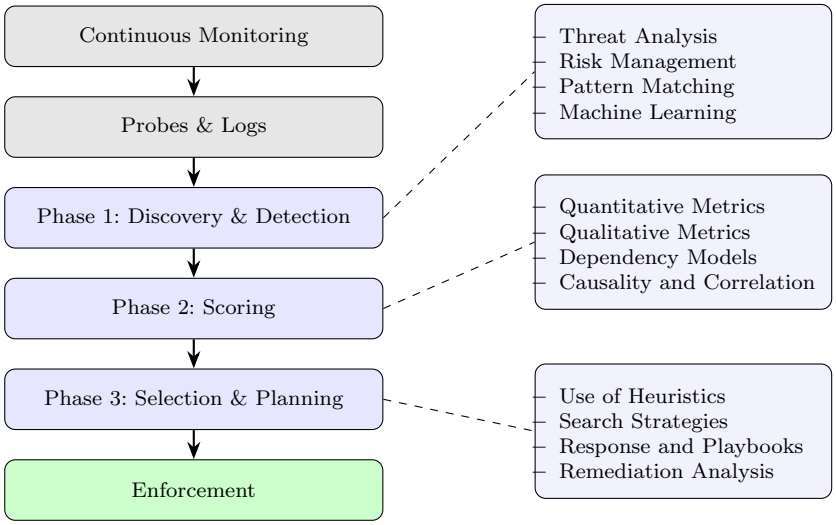


Fig. 5: Expanded phase-based framework, showing representative techniques associated with each stage.

We found that end-to-end automation systems, though robust, are static in their definition. Modular automation, on the other hand, offers much higher flexibility when considering the three phases of automation. To highlight this flexibility and build on the framework presented in Figure 5, we present Table 4. The table presents representative modular automation publications, together with the main technologies suggested in each publication, and the compatibility of these approaches as presented in Section 4.2.

Table 4 categorizes the phases of modular automation. It also shows the various stages of automation, namely discovery, scoring, and selection. The table is presented to emphasize the phase distribution of each surveyed work aiding the integration of multiple approaches. Based on the previous table, we establish a study of modulation while considering the three phases of automation.

4.3.1 Discovery and Detection

A robust approach can be identified in the combination of ML for anomaly detection. ML proves particularly effective when employed for detecting situations that deviate from normal conditions. We have identified two principal contributions that leverage ML within the realm of automation. In [37], the authors conduct anomaly detection utilizing a multilayer perceptron derived

Table 4: A summary table showing the technology (Tech.) used by publication and the compatibility of integration of each approach. The four technologies are Machine learning (ML), Decision Support (DS), Traditional (TR), and Novel (NV). The table highlights the compatibility score C and phases of automation (discovery, scoring, election) covered by each approach.

			Phase		
Approach	**Tech.**	**C**	**Discovery**	**Scoring**	**Selection**
Kilincer *et al.* [37]	ML	1.5	●		
Yang *et al.* [39]	ML	1.375	●		
Haque *et al.* [29]	TR	1.625	●		
Bringhenti *et al.* [36]	TR	1.625	●		
Coscia *et al.* [42]	TR	1.5	●		
Basak *et al.* [40]	ML	1.5	●		
Motzek *et al.* [22]	OM	0.625		●	●
Gonzalez *et al.* [27]	OM	0.75		●	●
Yaqoob *et al.* [26]	OM	0.5		●	
Shameli *et al.* [24]	DS	0.75		●	●
Soikkeli *et al.* [25]	DS	0.625		●	●
Sawik [34]	OM	0.625		●	
Ameri *et al.* [35]	TR	1.5		●	
Phillips *et al.* [41]	TR	2.5	●	●	
Paul *et al.* [31]	OM	0.625			●
Li *et al.* [28]	DS	0.75			●
Kavallieratos *et al.* [30]	TR	0.625		●	●
Martinelli *et al.* [23]	OM	0.625			●
Uuganbayar *et al.* [33]	OM	0.625			●
Stan *et al.* [32]	OM	0.75		●	●
Settanni *et al.* [38]	TR	1.75			●

from medical data. This method is primarily limited to detection, with no avenues for further research. In paper [39], ML is employed to automate the configuration of a general-purpose intrusion detection system. For instance, regular data can be converted into image streams and processed for the detection of malware instances [40]. The use of ML proves to hold robust and adaptable capabilities. However, training and customization of the precise ML algorithm remain the main challenge of such approaches. Other limitations include the large number of false positives that ML approaches may generate.

Some of the issues mentioned above can be combined with pre-existing solutions, such as enrichment of access control rule configurations [36] and use of pattern-matching techniques [42]. The integration of these solutions can be further improved with threat intelligence sharing platforms [29]. Utilizing established solutions provides the advantage of robust organizational backing. Conversely, these systems are predominantly static, and the scope for customization depends on the developers of these solutions.

4.3.2 Scoring

In addressing the challenge of scoring defense measures, one notable approach is the Return on Investment (ROI) framework. This method offers a foundational way to evaluate cybersecurity measures. It is further enhanced by the more comprehensive Financial and Operational RORI model [22]. The Stateful RORI model [27] adds customization features, making it suitable for attacker prediction frameworks. However, these models face key limitations. Their complexity grows rapidly in large systems, making them harder to manage. They also depend heavily on expert input during evaluation. In addition, a dedicated calculation phase is required to configure the RORI model and prepare the necessary system metrics. The ROSI calculation framework [26] presents a possible solution to this challenge. Still, a similar level of systematic approach is needed to automate the full RORI computation process.

Rather than examining the costs detailed in defense measures, researchers have investigated various metrics for scoring. The authors of [24] consider the historical and current impact of defense measures, with a particular focus on service quality and performance metrics. Progressing to [25], the researchers integrate attack and dependency graphs to monitor attack stages through attack path prediction. Nevertheless, they overlook the correlation among multiple defense measures. In the domain of supply chains, [34] enhances system security and reduces vulnerabilities by incorporating risk propagation into an elementary mathematical model. This approach fails to account for cost considerations or provide a holistic security blueprint. The three discussed approaches afford system managers increased flexibility when the primary concern for defense is not financial costs. Furthermore, these approaches may be collectively employed within a more comprehensive, albeit more complex, scoring framework for defense measures.

4.3.3 Selection and Planning

In this stage, approaches rely on previously introduced metrics, combined with optimization techniques, to determine an eventual set of defense measures to be enforced upon the system. A substantial contribution in this area is the Two-Stage Risk Assessment model, which uses government intelligence to estimate detection and containment costs [31]. However, this approach treats the system as a single entity and does not account for indirect risks, limiting its broader applicability. An alternative approach relies upon the application of a multi-path, NP-hard optimization task, guided by the system's attack surface [28]. A probabilistic attack-response tree, allowing for path optimization but introducing system challenges related to cross-correlation and scalability, is proposed. Similarly, risk assessment techniques can be combined with natural selection metaheuristics to select and prioritize defense measures [30].

A different line of work explores non-traditional approaches for defense measures selection. Instead of focusing solely on deterrence and mitigation, some studies integrate cybersecurity measures with insurance mechanisms [23].

This protection-insurance model is later enhanced using attack path modeling, from first-order logic to optimized security assessments [33]. While this approach adds flexibility, it still faces limitations related to risk propagation, mediated attacks, and system scalability.

Further contributions frame defense measures selection as a pathfinding problem [32], leveraging known NP-hard formulations to apply well-established heuristics. Another study examines vulnerabilities and attacks at a micro level [38]. In this approach, defense measures are treated as configurable security controls, organized like a recipe book to guide targeted interventions. This method is effective against simpler attack vectors, but its performance declines against more advanced threats. Together, these four approaches offer viable alternatives to conventional selection methods. They simplify the problem space by integrating insurance, reframing the optimization model, or narrowing the analysis to more manageable scopes.

To further highlight the concept of modulating automation of cybersecurity, we introduce the following end-to-end automation example. The example is built using multiple modular approaches:

- Bringhenti *et al.* [36] work is used to create a firewall setup that forms an initial protection for the system. Alongside, we use the Yang [39] IDS system to identify any threats that bypass the firewall.
- Gonzalez *et al.* [27] state RORI is used to produce a score for the defense measures that fit the detected vulnerabilities.
- Martinelli *et al.* [23] contribution can be used to form an insurance-oriented setup to compensate for any losses.

The previous example is one of the many possibilities that can be used to form end-to-end automation using modular approaches.

4.4 Analytical Look from the Surveyed Approaches

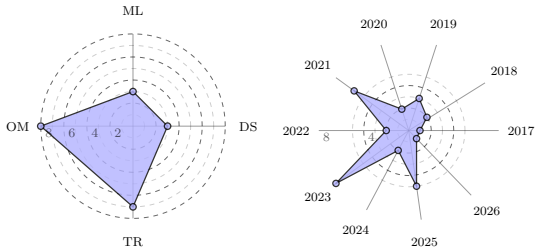
To understand the development of modular automation, we conducted an analytical study focusing on publication frequencies, temporal trends, and the underlying technological paradigms. By examining the temporal distribution of the selected works, two main trends emerge. First, there is a rapidly growing interest in the discovery phase, heavily driven by the integration of ML to automate data-driven diagnostic tasks. Second, the volume of end-to-end automation contributions remains low and inconsistent across the timeline. This scarcity is indicative of the complexity and long development cycles required to realize fully autonomous pipelines.

Further insights can be extracted by analyzing the technologies leveraged to develop these approaches. We categorize these methodologies into four distinct paradigms, as mapped out in Figure 6a:

- **Traditional (TR):** Comprises foundational, deterministic security mechanisms, including but not limited to IDS, firewalls, and signature-based mitigation frameworks.

- **Decision Support (DS):** Encompasses formal graphical and logical tools—such as attack graphs, defense trees, and deterministic decision structures—designed to model threat propagation.
- **Machine Learning (ML):** Consists of data-driven, probabilistic frameworks leveraging statistical learning models to discover anomalies or automate pattern recognition tasks.
- **Advanced Optimization & Cross-Domain Modeling (OM):** Represents advanced paradigms that reshape cybersecurity challenges into alternative domains, specifically leveraging operations research, formal mathematical optimization, or transformations (e.g., computer vision).

The technological mapping of each surveyed contribution is detailed in Table 4. Studying the time map, Figure 6b illustrates a sustained and steady academic interest between 2010 and 2025. Concurrently, Figure 6a highlights how modern frameworks increasingly blend traditional logic with advanced optimization frameworks. This heterogeneous distribution underscores an adaptability within modern cybersecurity automation. Granular insights derived from an individual examination of each technological paradigm are detailed below.



a Technologies utilized in publication.

b Publications throughout the years.

Fig. 6: Radar showing the distribution of publications across the four main technology types and the years. The four technologies are Machine learning (ML), Decision Support (DS), Traditional (TR), and Novel (NV).

4.4.1 Traditional

Traditional approaches continue to demonstrate foundational efficiency across the defensive spectrum. These solutions are anchored upon well-established software subsystems that have been refined over decades to address acute operational threats. To maintain relevance and integration, approaches typically incorporate these subsystems with minor API or middleware adjustments. However, the primary limitation of these technologies stems from the highly dynamic and polymorphic nature of modern cyber threats. Consequently, a

pivot has emerged, where researchers pair traditional approaches with ML components to close the adaptability gap [36].

4.4.2 Decision Support

Decision support mechanisms excel at formalizing defensive logic, scoring system vulnerabilities, and mapping defensive actions, but exhibit limitations in threat discovery. Outside of data-driven ML integrations, decision frameworks offer minimal utility for discovery, a gap primarily driven by seven distinct cyber-physical and network constraints:

- Strict resource and computational bottlenecks inherent to legacy CPS hardware [53].
- The highly dynamic, volatile nature of modern network topologies [54].
- Massive traffic volumes paired with a lack of high-fidelity labeled datasets [55].
- An expanding attack landscape characterized by highly sophisticated, state-sponsored evasion techniques [56].
- A historic lack of protocol standardization across heterogeneous deployments [57].
- The systemic reality that legacy infrastructure was fundamentally engineered without native security primitives [58].
- An operational dependency on data pools that frequently obscure non-linear attack characteristics.

4.4.3 Machine Learning

While ML represents a highly robust and evolving paradigm, its operational footprint within the surveyed literature remains largely confined to the discovery and detection. ML models excel at parsing complex, high-dimensional configurations for pattern recognition. Conversely, countermeasure quantification, resource allocation, and strategic selection, becomes significantly less viable. In these cases, exact mathematical modeling and closed-form optimizations yield vastly superior complexity and accuracy. Furthermore, introducing ML into critical pipelines injects severe vulnerabilities regarding model robustness. These vulnerabilities include adversarial exploits—such as input manipulation, data poisoning, and model inversion attacks [59].

4.4.4 Advanced Optimization & Cross-Domain Modeling

This category represents the most dominant and mathematically rigorous trend with a shift toward engineering abstractions in cybersecurity. Innovation in this space falls into two main methodologies: mathematical optimization and cross-disciplinary translation. The first sub-trend leverages formal mathematical and algorithmic frameworks—such as two-stage stochastic programming to optimize resource allocation [32] or heuristic pathfinding models [39]. These strategies yield mathematical solutions that preserve robustness and scalability. The second sub-trend relies on cross-domain structural mapping, such as

translating network log configurations into image processing matrices to leverage advanced spatial pattern recognition models [40]. By abstracting cybersecurity problems into mature, alternative fields of science, these approaches expand the defensive toolkit. These optimization engines can solve highly complex, multi-objective security challenges that traditional frameworks cannot elegantly address.

4.5 Strengths and Weaknesses

This section forms a set of strengths and weaknesses for the surveyed publications. We present the results in Tables 5, 6. In the tables, we show the external data, if used, the strengths, and the weaknesses of each. None signifies no external data is needed, and unspecified is when the source is not specified. We evaluate each article based on the survey we conducted. Specifically, we compare these publications and build a set of benchmarks. We found that though some publications have weaknesses in one aspect, they introduce innovative ideas in others.

Table 5: Automation approaches for Attack Detection and Classification. This table presents a comparative overview of methods centered on detecting cyber threats. The data was compiled through an in-depth literature review.

Approach	**Data Type**	**Technical Strengths**	**Limitations**
Kilincer *et al.* [37]	Network & Telemetry Data	Robust RFE feature reduction. Tested on 4 IoMT datasets.	Exhaustive grid search. Requires periodic retraining.
Yang *et al.* [39]	Network Features	Automated AutoML pipeline. TVAE synthesis handles imbalances.	Requires quality tabular features. Complex stacking ensemble.
Basak *et al.* [40]	Malware Binaries (Images)	Automated features extraction. Captures accurate spatial indicators.	Weak obfuscation variants. High memory overhead.
Haque *et al.* [29]	Structured CTI (STIX/TAXII)	Secure, model threat sharing. ReBAC policies.	Requires threat indicators. Low scalability due to ReBAC.
Bringhenti *et al.* [36]	Topologies & Security Intents	Standardized firewall setup. Min rule set and allocation overhead.	Blind to payload anomalies. Constraint solvers Low scalability.
Coscia *et al.* [42]	API Call Sequences	YARA rule automation. 9 benchmark datasets.	Blind to malware hiding outside API boundaries.

The tables show a persistent challenge of dependence on external data. For instance, the approach outlined in [19] utilizes CVSS, while [31] depends on governmental datasets. Although leveraging pre-existing databases offers convenience, it simultaneously poses a potential vulnerability. Specifically, these data sets may lack coverage of novel or emerging attack scenarios, leaving the system susceptible to risks. This dependency necessitates ongoing updates to the database to integrate new threat information. Should the defender's

system rely on the occurrence of attacks to initiate defense, it establishes a substantial vulnerability. When sensitive data is compromised, the need to protect the system is no longer necessary.

Conversely, methodologies like the Stateful RORI [27] thoroughly examine the influence of defense measures; however, they are encumbered by the high complexity of their algorithms. Within this framework, external data is employed to address known attacks. In contrast, the remaining novel attacks are addressed in real-time through methodologies such as attack graphs and RORI.

A similar challenge is seen in the scalability of approaches that rely on attack graphs or genetic heuristics. These approaches often struggle to handle larger systems without incurring exponential complexity.

Risk assessment models illustrate the challenge of balancing complexity, adaptability, and scalability. Models like the Biological Security Immune System [19] and Knowledge-Based Cybersecurity Risk Assessment [41] provide rich and adaptable frameworks. Conversely, these models require significant customization and prior knowledge, making them less suitable for rapidly changing threat landscapes. Meanwhile, simpler models, such as Budget Management in Supply Chains [34], prioritize efficiency but may sacrifice depth and cross-correlation. This sacrifice limits the model's effectiveness in complex environments.

5 Open Research Directions and Challenges

While cybersecurity automation has progressed significantly across isolated modules, achieving unified, resilient, and enterprise-ready automation pipelines remains an open challenge. This section synthesizes systemic gaps identified during our literature review and highlights critical research priorities required to transition modular automation from theoretical frameworks to deployment realities.

5.1 Bridging Strategy Quantification and Real-Time Enforcement

A disconnect remains between defense measures evaluation models and real-time execution layers. Current automation pipelines heavily rely on static, scoring-based metrics (e.g., CVSS vectors) or historical government databases to determine defensive responses, as observed by Paul *et al.* [31]. However, these frameworks struggle to capture the transient dynamics, multi-step propagation, and long-term operational impacts of modern attacks.

Future research must prioritize the development of adaptive, continuous evaluation loops capable of weighing mitigation against long-term costs. Transitioning from discrete scoring systems to continuous, context-aware risk estimation engines is vital for ensuring that automated responses do not introduce unforeseen self-inflicted denial-of-service conditions.

Table 6: Automation approaches for Response Selection and Scoring. This table presents a comparative overview of methods centered on detecting cyber threats. The data was compiled through an in-depth literature review.

Approach	Data Type	Technical Strengths	Limitations
Motzek *et al.* [22]	Mission profiles & Asset values	RORI and operational metrics. Pareto-efficient mitigation strategies.	Low scalability in asset dense networks
Gonzalez *et al.* [27]	Historical state logs & System topologies	Temporal security dependencies. Hypergraphs conflict solving.	Demands a highly precise, low-latency estimation of parameters.
Yaqoob *et al.* [26]	CVSS metrics & Network configurations	Bayesian probability forming of ROSI. Standard threat indices.	Restricted to static data. Struggles with zero-day attacks.
Shameli *et al.* [24]	ADTs & Service dependency metrics	Adapts to real-time & past metrics. Models dependencies using SDGs.	Constructing Attack defense trees for large networks is complex.
Soikkeli *et al.* [25]	Vulnerability maps & Action logs	Mitigation, uptime, and recovery cost. Balances inaction and containment.	Requires knowledge of service propagation delays.
Sawik *et al.* [34]	Multi-tier supplier risk databases	Targets supply chain weak points. Direct and indirect loss vectors.	Linearized bounds can oversimplify non-linear operational bottlenecks.
Ameri *et al.* [35]	Heterogeneous ICS vendor manuals	Automated requirement extraction. Vendor specific Configurations.	Dependent on document quality. No model control deployment costs.
Phillips *et al.* [41]	Ontology classes & Expert rulesets	Systematic cause-and-effect study. Exposes indirect risk paths.	Requires comprehensive understanding of threat knowledgebase.
		Selection	
Paul *et al.* [31]	CTI & Gov. Data	Minimizes social cost. Accounts for investment.	Dependent on CTI feeds. Lacks risk propagation.
Li *et al.* [28]	Multi-path Metrics	Evaluates impacts dynamically. Optimizes multi-path response.	Extensive pre-calibration. Low scalability tree models.
Kavallieratos *et al.* [30]	Risk Vectors	Captures complex risk propagation. Multi-objective optimization.	Subjective threat models. Genetic slow convergence rates.
Martinelli *et al.* [23]	Insurance Data	Bridges finance and tech controls. Independent of detection.	Simplified linear assumptions. Fits static, budget-constrained setups.
Uuganbayar *et al.* [33]	Insurance Data	Non-linear insurance premium. Metaheuristic utility scaling.	Assumes uniform asset dependencies and static risk pools.
Stan *et al.* [32]	CVSS Framework & Attack Graphs	Maps vulnerabilities to mitigations. Efficient A* pathfinding search.	Converting graphs to pathfinding chains can lose dimensionality.
Settanni *et al.* [38]	Ontology & Capability Repositories	Vendor-neutral abstract recipes. Executable CACAO playbooks.	Dependent on a recipe book. No cost-benefit metrics.

5.2 Reproducibility, Open Science, and Long-Term Lifecycle Support

As established in Section 4.2 and by Alimi *et al.* [60], a vast majority of the surveyed work lack source code and deployment manifests. This lack of reproducibility slows down research, preventing the community from establishing standard baselines to benchmark new methodologies.

Fortunately, this deficit is easily addressable, given that the network datasets are largely public or synthesizable. Furthermore, research must address the total absence of long-term lifecycle support within academic prototypes. There is a need for modular components that can seamlessly adapt to network changes without requiring architectural redesigns.

5.3 Standardization, Orchestration, and Modular Synthesizability

The multi-tiered modular automation proposed in Section 4.3 remains largely theoretical. Current literature frequently treats security modules—such as alert systems, root-cause diagnostics, and mitigation selection—as independent subsystems. Consequently, there is a research gap regarding the integration, orchestration, and communication syntax required to bind these heterogeneous components into an ecosystem. Future research should focus on building standardized API middleware capable of handling data across varied execution environments. These middlewares require conflict-resolution protocols for when multiple automated modules execute overlapping or contradictory security policies.

5.4 Scalability and Complexity Management in System Modeling

Industrial adoption of automation frameworks is bottle-necked by network complexity. While formal structures such as attack trees, defense graphs, and extensions of frameworks like MulVAL [20] offer rigorous analytical foundations for modeling vulnerabilities [61], their computational complexity scales exponentially. To bridge this deployment gap, future studies must design abstraction layers capable of dynamically optimizing graph-based models without compromising security. Research should investigate clustered modeling techniques that isolate sub-networks while preserving macro-level accuracy.

5.5 Robust Game-Theoretic Frameworks Under Deep Uncertainty

To transition from reactive to proactive approaches, requires modeling that minimize maintenance and defense costs [62]. Recent paradigms have successfully framed these interactions using non-zero-sum, two-player Stackelberg security games, including networked control systems [63], cloud infrastructures [64], and wireless communication topologies under DoS constraints [65].

Despite these theoretical advancements, existing game-theoretic models remain constrained by an assumption of perfect attacker profiles or stationary environmental states. A major open research gap lies in engineering robust, decision-making algorithms that compute game-theoretic equilibria under conditions of uncertainty, incomplete threat intelligence, and asymmetric resource constraints. Enhancing these frameworks with real-time parameter will allow proactive defense systems to dynamically handle sophisticated and adaptive adversaries.

6 Conclusion

The automation of cybersecurity in CPS systems presents significant challenges due to the complexity of attacks and the demanding nature of system management. This study surveyed various defense solutions and automation approaches, proposing a three-phase framework: discovery, scoring, and selection. Each contribution surveyed was classified according to its primary focus, limitations, and the results were analyzed. Further classification considers the extent of automation where approaches are considered either to be modular or end-to-end. Modular approaches cover two or fewer of the three automation phases. End-to-end approaches cover the three phases of automation and can function as an all-in-one solution. Finally, the survey introduces an index of compatibility built on eight criteria. The compatibility metric provides an index allowing for easier integration of multiple modular approaches.

The findings reveal that the literature is rich with multiple techniques and methodologies for handling cybersecurity automation. In addition to automation-focused strategies, researchers have explored various recovery-oriented ones, such as those that incorporate cyber insurance and cost optimization models. In doing so, the focus is on mitigating the financial impact of attacks rather than solely preventing or defending against them. Strategies span preemptive, damage mitigation, and insurance-based models. This diversity presents the opportunity to combine multiple approaches for a more robust and comprehensive cybersecurity solution.

Acknowledgements

This work was supported in part by the Spanish Government through the "SECURING" Project from the Ministerio de Ciencia e Innovación, the Agencia Estatal de Investigación, and the European Regional Development Fund (ERDF) under Grant PID2021-125962OB-C31; the ARTEMISA International Chair of Cybersecurity and the DANGER Strategic Project of Cybersecurity, both funded by the Spanish National Cybersecurity Institute (INCIBE) through the European Union—NextGenerationEU and the Recovery, Transformation and Resilience Plan. Authors would like to acknowledge as well support from the International Alliance for Strengthening Cybersecurity and Privacy in Healthcare (CybAlliance, Project no. 337316).

References

1. K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (idps). Tech. rep., NIST (2007)
2. R. Sommer, V. Paxson, in *2010 IEEE symposium on security and privacy* (IEEE, 2010), pp. 305–316
3. S. García, M. Grill, J. Stiborek, A. Zunino, Computers & Security **45**, 100 (2014). DOI 10.1016/j.cose.2014.05.011
4. J. Pinto. Automation Systems Cyber Security (2003). URL `https://www.automation.com/en-us/articles/2003-1/automation-systems-cyber-security`
5. S. Sinha. State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally (2024). URL `https://iot-analytics.com/number-connected-iot-devices/`. Accessed: 2025-08-31
6. Trendmicro. Understanding Targeted Attacks: Goals and Motives - Security News (2015). URL `https://www.trendmicro.com/vinfo/us/security/news/cyberattacks/understanding-targeted-attacks-goals-and-motives`. Accessed: 2025-08-31
7. A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, *Handbook of applied cryptography* (CRC press, 2018)
8. M. Bishop, *Introduction to computer security* (Addison-Wesley Professional, 2004)
9. S. Dharmapurikar, J.W. Lockwood, IEEE Journal on Selected Areas in communications **24**(10), 1781 (2006)
10. E. Al-Shaer, Q. Duan, J.H. Jafarian, in *International Conference on Security and Privacy in Communication Systems* (Springer, 2012), pp. 310–327
11. E. Casey, *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet* (Academic Press, 2011)
12. T.D. Le, T. Le-Dinh, S. Uwizeyemungu, Electronics **14**(11), 2252 (2025). DOI 10.3390/electronics14112252
13. P. Nespoli, D. Papamartzivanos, F. Gómez Mármol, G. Kambourakis, IEEE Communications Surveys Tutorials **20**(2), 1361 (2018). DOI 10.1109/COMST.2017.2781126
14. ACM. ACM Digital Library: Computing Research and Publications (2025). URL `https://dl.acm.org`. Accessed: 2025-08-31
15. IEEE. IEEE Xplore: Digital Library for Engineering and Technology (2025). URL `https://ieeexplore.ieee.org`. Accessed: 2025-08-31
16. Scopus. Scopus: Abstract and citation database (2025). URL `https://www.scopus.com`. Accessed: 2025-08-31
17. N. Poolsappasit, R. Dewri, I. Ray, IEEE Transactions on Dependable and Secure Computing **9**(1), 61 (2012). DOI 10.1109/TDSC.2011.34
18. B. Kordy, S. Mauw, S. Radomirović, P. Schweitzer, Journal of Logic and Computation **24** (2014). DOI 10.1093/logcom/exs029
19. P. Nespoli, F.G. Mármol, J.M. Vidal, IEEE Access **9**, 60971 (2021). DOI 10.1109/ACCESS.2021.3074021

20. O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, A. Shabtai, IEEE Transactions on Dependable and Secure Computing **19**(3), 1936 (2022). DOI 10.1109/TDSC.2020.3041999
21. L. Alevizos, International Journal of Information Technology **17**(2), 767 (2025). DOI 10.1007/s41870-024-02324-9
22. A. Motzek, G. Granadillo, H. Debar, J. Garcia-Alfaro, R. Möller, EURASIP Journal on Information Security **2017** (2017). DOI 10.1186/s13635-017-0063-6
23. F. Martinelli, G. Uuganbayar, A. Yautsiukhin, in *ICT Systems Security and Privacy Protection*, ed. by L.J. Janczewski, M. Kutyłowski (Springer International Publishing, Cham, 2018), pp. 187–200. DOI 10.1007/978-3-319-99828-2_14. URL `https://link.springer.com/chapter/10.1007/978-3-319-99828-2_14`
24. A. Shameli-Sendi, H. Louafi, W. He, M. Cheriet, IEEE Transactions on Dependable and Secure Computing **15**(5), 755 (2018). DOI 10.1109/TDSC.2016.2615622
25. J. Soikkeli, L. Muñoz-González, E. Lupu, in *Proceedings of the 14th international conference on availability, reliability and security* (2019), pp. 1–10
26. T. Yaqoob, A. Arshad, M.F. Abbas, H.and Amjad, N. Shafqat, Future Generation Computer Systems **95**, 754 (2019). DOI https://doi.org/10.1016/j.future.2018.12.033
27. G. Gonzalez-Granadillo, E. Doynikova, J. Garcia-Alfaro, I. Kotenko, A. Fedorchenko, Journal of Information Security and Applications **54**, 102562 (2020). DOI https://doi.org/10.1016/j.jisa.2020.102562
28. F. Li, Y. Li, S. Leng, Y. Guo, K. Geng, Z. Wang, L. Fang, Computers & Security **97**, 101927 (2020). DOI https://doi.org/10.1016/j.cose.2020.101927
29. M.F. Haque, R. Krishnan, Information Systems Frontiers **23**(4), 883 (2021). DOI 10.1007/s10796-020-10103-7
30. G. Kavallieratos, G. Spathoulas, S. Katsikas, Sensors **21**(5) (2021). DOI 10.3390/s21051691
31. J.A. Paul, M. Zhang, European Journal of Operational Research **291**(1), 349 (2021). DOI https://doi.org/10.1016/j.ejor.2020.09.013
32. O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, A. Shabtai, in *2021 IEEE 34th Computer Security Foundations Symposium (CSF)* (2021), pp. 1–16. DOI 10.1109/CSF51468.2021.00003
33. G. Uuganbayar, A. Yautsiukhin, F. Martinelli, F. Massacci, Computers & Security **101**, 102121 (2021). DOI https://doi.org/10.1016/j.cose.2020.102121
34. T. Sawik, International Journal of Production Research **60**(2) (2022). DOI 10.1080/00207543.2021.1914356
35. K. Ameri, M. Hempel, H. Sharif, J. Lopez, K. Perumalla, ACM Trans. Manage. Inf. Syst. **14**(1) (2023). DOI 10.1145/3546580
36. D. Bringhenti, R. Sisto, F. Valenza, in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)* (2023), pp. 293–295. DOI

10.1109/NetSoft57336.2023.10175442
37. I. Firat Kilincer, F. Ertam, A. Sengur, R. Tan, U. Rajendra Acharya, Biocybernetics and Biomedical Engineering **43**(1), 30 (2023). DOI https://doi.org/10.1016/j.bbe.2022.11.005
38. F. Settanni, L. Regano, C. Basile, A. Lioy, in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)* (2023), pp. 492–497. DOI 10.1109/NetSoft57336.2023.10175486
39. L. Yang, A. Shami, in *Proceedings of the Workshop on Autonomous Cybersecurity* (2023), pp. 68–78. DOI 10.1145/3689933.3690833
40. M. Basak, M.M. Han, Sensors **24**(11), 3406 (2024). DOI 10.3390/s24113406
41. S.C. Phillips, S. Taylor, M. Boniface, S. Modafferi, M. Surridge, IEEE Access **12**, 82482 (2024). DOI 10.1109/ACCESS.2024.3404264
42. A. Coscia, R. Lorusso, A. Maci, G. Urbano, Computers & Security **153**, 104397 (2025). DOI 10.1016/j.cose.2025.104397
43. B. Schneier, Dr. Dobb's journal **24**(12), 21 (1999)
44. B. Xu, Z. Zhong, G. He, Security and Communication Networks **2020**, 8870734 (2020). DOI 10.1155/2020/8870734
45. X. Ou, S. Govindavajhala, A.W. Appel, in *USENIX security symposium*, vol. 8 (Baltimore, MD, 2005), vol. 8, pp. 113–128. URL `https://www.usenix.org/event/sec05/tech/full_papers/ou/ou_html`
46. P. Nespoli, F. Gomez Marmol, J. Maestre Vidal, Cluster Computing **24**, 1 (2021). DOI 10.1007/s10586-020-03198-9
47. National Institute of Standards and Technology. Cybersecurity framework. URL `https://www.nist.gov/cyberframework`
48. International Organization for Standardization. ISO/IEC 27001:2022 (2025). URL `https://www.iso.org/standard/27001`
49. The MITRE Corporation. Campaigns | MITRE ATT&CK® (2025). URL `https://attack.mitre.org/campaigns/`. Accessed: 2025-08-31
50. M. Jeffrey, Internet Encyclopedia **3**, 211 (2004)
51. N. Kheir, N. Cuppens-Boulahia, F. Cuppens, H. Debar, in *Computer Security – ESORICS 2010*, ed. by D. Gritzalis, B. Preneel, M. Theoharidou (Springer, 2010), Lecture Notes in Computer Science, pp. 626–642. DOI 10.1007/978-3-642-15497-3_38
52. W. Sonnenreich, J. Albanese, B. Stout, Journal of Research and Practice in IT **38**(1), 45 (2016). DOI 10.3316/ielapa.937199632104879
53. R. Roman, J. Zhou, J. Lopez, Computer Networks **57**(10), 2266 (2013). DOI 10.1016/j.comnet.2012.12.018
54. M.M. Hossain, M. Fotouhi, R. Hasan, in *2015 IEEE World Congress on Services* (2015), pp. 21–28. DOI 10.1109/SERVICES.2015.12
55. Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N.O. Tippenhauer, J.D. Guarnizo, Y. Elovici. Detection of unauthorized IoT devices using machine learning techniques (2017). DOI 10.48550/arXiv.1709.04647
56. I. Butun, P. Österberg, H. Song, IEEE Communications Surveys & Tutorials **22**(1), 616 (2020). DOI 10.1109/COMST.2019.2953364

57. S. Sicari, A. Rizzardi, L.A. Grieco, A. Coen-Porisini, Computer Networks **76**, 146 (2015). DOI 10.1016/j.comnet.2014.11.008
58. P. Williams, I.K. Dutta, H. Daoud, M. Bayoumi, Internet of Things **19**, 100564 (2022). DOI 10.1016/j.iot.2022.100564
59. K. Hartmann, C. Steup, in *2020 12th International Conference on Cyber Conflict (CyCon)*, vol. 1300 (2020), vol. 1300, pp. 327–349. DOI 10.23919/CyCon49761.2020.9131724
60. K.O. Adefemi Alimi, K. Ouahada, A.M. Abu-Mahfouz, S. Rimer, Sensors **20**(20), 5800 (2020). DOI 10.3390/s20205800
61. H.S. Lallie, K. Debattista, J. Bal, Computer Science Review **35**, 100219 (2020). DOI https://doi.org/10.1016/j.cosrev.2019.100219
62. F. Mizrak, Pressacademia (2023). DOI 10.17261/Pressacademia.2023.1807
63. P. Shukla, L. An, A. Chakrabortty, A. Duel-Hallen, IEEE Transactions on Control Systems Technology **31**(2), 856 (2023). DOI https://doi.org/10.1109/TCST.2022.3207671
64. A. Ait Temghart, M. Marwan, M. Baslam, Security and Communication Networks **2023**(1), 2811038 (2023). DOI https://doi.org/10.1155/2023/2811038
65. Z. Wang, H. Shen, H. Zhang, S. Gao, H. Yan, Information Sciences **642**, 119134 (2023). DOI https://doi.org/10.1016/j.ins.2023.119134