IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 April 2023, accepted 19 April 2023, date of publication 1 May 2023, date of current version 10 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272053

TOPICAL REVIEW

A Review on Attack Graph Analysis for IoT Vulnerability Assessment: Challenges, Open Issues, and Future Directions

OMAR SAIF MUSABBEH BIN HAMED ALMAZROUEI[1], PRITHEEGA MAGALINGAM[1], MOHAMMAD KAMRUL HASAN[2], (Senior Member, IEEE), AND MOHANA SHANMUGAM[3]

[1]Advanced Informatics Department, Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
[2]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[3]College of Computing and Informatics, Universiti Tenaga Nasional, Kajang 43000, Malaysia

Corresponding author: Pritheega Magalingam (mpritheega.kl@utm.my)

This work was supported by the Ministry of Higher Education, Government of Malaysia, and the Research Management Centre, Universiti Teknologi Malaysia, through the Fundamental Research Grant Scheme (FRGS) Grant (vote number R.K130000.7856.5F016) under Grant FRGS/1/2018/ICT04/UTM/02/8.

ABSTRACT Vulnerability assessment in industrial IoT networks is critical due to the evolving nature of the domain and the increasing complexity of security threats. This study aims to address the existing gaps in the literature by conducting a comprehensive survey on the use of attack graphs for vulnerability assessment in IoT networks. Attack graphs serve as a valuable cybersecurity tool for modeling and analyzing potential attack scenarios on systems, networks, or applications. The survey covers the research conducted between 2016 and 2021(34 peer-reviewed journal articles and 28 conference papers), identifying and categorizing the main methodologies and technologies employed in generating and analyzing attack graphs. In this review, core modeling techniques for IoT vulnerability assessment are highlighted, such as Markov Decision Processes (MDP), Feature Pyramid Networks (FPN), K-means clustering, and logistic regression models, along with other techniques involving genetic algorithms like fast-forward (FF), contingent fast-forwards (CFF), advanced reinforcement-learning algorithms, and HARMs models. The evaluation of the performance of these attack graph models using IoT networks or devices as case studies is also emphasized. This survey provides valuable insights into the state-of-the-art attack graph techniques for IoT network vulnerability assessment, identifying various applications, performances, research opportunities, and challenges. As a reference source, it serves to inform academicians and practitioners interested in leveraging attack graphs for IoT network vulnerability assessment and guides future research directions in this area.

INDEX TERMS Attack graph, the Internet of Things, network vulnerabilities, vulnerability assessment.

I. INTRODUCTION

The era of hyper-intelligence, hyper-convergence, and hyper-connectivity established by the Industry 4.0 revolution continues in earnest as the industrial Internet of Things (IoT) devices and environments develop. The IoT creates a new paradigm for industrial networking where sensors, actuators, and network devices become crucial elements for industrial communications [1]. To this end, various devices may be considered ‘‘smart’’ since they contain network transceivers and microprocessors, facilitating communication and allowing autonomous services. Consequently, IoT is a promising research field related to developing devices connected to the World Wide Web and promoting smart environments. Furthermore, technological advances and communication have created a vastly connected world [2]. Many mundane IoT devices are connected to enterprise and private networks, including smart bulbs, baby monitors, smart cameras, smart televisions, and smart vacuum cleaners. This has made networks easy targets for attackers and fraudsters, who can easily conceal their fraudulent activities within the volumes of data [3]. With networks growing exponentially, the

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

opportunities for attackers to manipulate them for personal benefits have expanded significantly [2].

Most organizations have invested hugely, including adopting sophisticated mechanisms and innovative technologies, to secure their data and networks from external and internal threats [4]. Specifically, much focus has been directed on analyzing the activities and interactions of users and customers within a network. Despite the investments, detecting system breaches in the current big data environment are akin to finding a needle in a haystack. Rather than focusing on reacting to system breaches after they have already happened, much attention has turned to vulnerability identification. Vulnerability of IoT networks refers to security weaknesses or flaws in the various interconnected devices, sensors, and networks that make up the IoT ecosystem [36]. An attacker can exploit these vulnerabilities to gain unauthorized access to IoT devices, manipulate data, steal sensitive information, or disrupt critical infrastructure. Several types of vulnerabilities can be present in an IoT network, some of which are weak or default passwords, lack of data encryption, outdated software in IoT devices, insecure web or mobile interfaces, lack of network segmentation, malware and botnets [36].

Vulnerability assessment in the literature includes various activities that aim to inform or improve mitigation strategies for a network or a system [5]. This is achieved by systematically reviewing security weaknesses in the network through information gathering. Vulnerability assessment aims to establish whether the network (or system) is predisposed to any known vulnerabilities before assigning severity levels to the identified vulnerabilities and recommending mitigation or remediation if and whenever necessary [6]. Thus, vulnerability assessment is among the most effective approaches for recommending ways to strengthen the target network's security level [3]. While vulnerability assessment is a favored approach, the process requires significant time and extensive financial resources. Consequently, it is becoming increasingly challenging to perform vulnerability assessments because of the complexity of modern networks and the heightened information systems security [6]. Hardware and software developers have increased security awareness, but vulnerability assessment models are fragmented, making the concept challenging [5].

Attack graphs reveal all potential combinations of vulnerabilities and their relationships, which are important for preventing multistep assaults. In other words, they expose potential dangers to networks by outlining all potential attack routes. Hence, it is a good technique for extracting paths on how to protect network nodes against innate vulnerabilities. A security analyst may find it difficult to identify which vulnerabilities should be fixed in an attack graph when an IoT system's device count and associated vulnerabilities grow. In order to effectively implement countermeasures, automatic extraction of recommendations from attack graphs and their user accessibility can be crucial.

Notably, while there have been state-of-the-art attack graph models and frameworks proposed for handling IoT vulnerability assessment activities [6], [7], [8], [12], there is a lack of a meta-analysis or systematic literature review of the existing literature. For example, Hydara et al. [8] and McKinnel et al. [9] performed systematic literature reviews on various aspects of vulnerability assessment. The evaluated literature contains extensive studies on the IoT architecture, protocols, developing technologies, IoT attacks, and dangers. However, no thorough study has addressed IoT vulnerabilities and their evaluation using attack graphs. Although some papers capture both attack graphs and IoT [3], [34], they either do not cover certain topics, such as the parameters of the IoT network used to develop the attack graph [8], [17], [35], [48], [52], [54], [57], [68], [69] and the methods and tools used for visualizing the model, framework, or application [6], [16], [27], [30], [37], [39], [40], [62], or are no longer fully relevant due to the rapidly evolving domain. This survey paper is needed because IoT systems are becoming increasingly complex and pervasive in our daily lives, making it crucial to ensure their security. A comprehensive understanding of IoT vulnerabilities and their evaluation using attack graphs can help researchers, practitioners, and decision-makers better assess and mitigate potential security risks in IoT systems. By consolidating and analyzing the current state of research on this topic, this survey paper provides valuable insights that can guide future research directions, inform the development of effective security solutions, and enable organizations to make informed decisions about securing their IoT infrastructure. In addition, this survey paper can serve as a reference point for those seeking to understand the intersection of IoT security and attack graph analysis, promoting the advancement of the field and the development of novel solutions. For instance, Gupta et al. have assembled in their survey the historical context of the IoT, meticulously researched the IoT's design, and varied types of issues it may encounter. Additionally, they considered the main problems and available fixes for permissive technologies like Radio-Frequency Identification (RFID) and Wireless Sensor Networks (WSN). Similarly, Atzori et al. investigated IoT in many scenarios, covered enabling technologies, and analyzed how they affected daily life. In this study, we have looked at a large number of these associated surveys to determine their contributions and show how the current study advances the state-of-the-art in IoT security.

After discussing the IoT networks and attack graphs, it is essential to provide a list of commonly used acronyms and their meanings for better understanding. Table 1 presents the acronyms used throughout this paper along with their explanations.

The exploration of attack graphs and IoT networks is a relatively new area of research. In light of the recent diverse developments in the IoT ecosystem, this paper presents a comprehensive survey and critical evaluation of the literature on the use of attack graphs for evaluating security weaknesses

TABLE 1. Acronyms and their meanings.

Acronym	Explanation
IoT	Internet of Things
IEEE	Institute of Electrical and Electronics Engineers
RFID	Radio-Frequency Identification
WSN	Wireless Sensor Networks
MDP	Markov Decision Process
POMDP	Partially Observable Markov Decision Process
SCADA	Supervisory Control and Data Acquisition (SCADA) Networks
HARM	Hierarchical Attack Representation Models
FPN	Feature Pyramid Networks

in IoT networks. The challenges were identified through a comprehensive review of the existing literature on the use of attack graphs for vulnerability assessment in IoT networks. This included a systematic analysis of peer-reviewed journal articles and conference papers on the topic, as well as a consideration of practical experiences and limitations in current methods. The authors conducted a thorough examination of the existing knowledge in the field and identified the gaps and limitations in current technologies and methodologies. Through this process, it is able to identify the following challenges in using attack graphs for detecting security weaknesses in contemporary IoT networks:

1) Technologies and methodologies that provide adequate representation of the IoT network parameters in the attack graph.
2) Technologies and methodologies for generating attack graphs in IoT networks, especially in networks with 1000 nodes or more.
3) Technologies and methodologies for attack graph analysis and the formulation of security properties and vulnerability detection.
4) Technologies and methodologies of attack graph visualization.
5) Technologies and methodologies for recommendation and the implementation of response strategies, where we evaluate the recent innovations that have significantly improved these tasks.

Thus, we seek to contribute to this research by analyzing the existing attack graph techniques utilized to find security weaknesses in IoT networks, emphasizing the different applications and their performance. Apart from the review methodology and findings, this paper shows the methodology for managing the graph's input and output and the technologies for attack graph generation and analysis. Finally, this paper concludes with potential opportunities for future research.

II. ORGANIZATION

The paper is organized as follows. Section III provides an overview of the Literature Review Method used in the study. Particularly, it discusses the step-by-step process taken for the systematic review. Section IV discusses the results obtained from the systematic review. The taxonomy of the IoT vulnerability assessment technique, specifically the attack graphs, is discussed in Sections V and VI. The following sections provide the recommendations and the paper's conclusion.

III. LITERATURE REVIEW METHOD

To find relevant studies, a set of search queries were made by combining the most important research terms, such as "vulnerability assessment," "attack graphs," and "security weaknesses". The search was restricted to three databases: IEEE Explore, ACM Digital Library, and ScienceDirect. This is because the three databases index the majority of journals and conference papers related to computer science and engineering. Further, the title, keywords, and abstract were assessed manually using different combinations of the research databases listed in Table 2 to identify the most relevant articles to the study. The literature search queries were completed for studies published between 2017 and 2021, and the metadata was included in the initial search.

Moreover, a snowballing process was used to search through the references of the located papers and to find additional relevant papers. This snowballing process was conducted for both forward and backward lookups and was completed once most papers related to the study were found. The inclusion and exclusion criteria were applied following this preliminary dataset construction to refine only the most relevant papers to our study (see Table 3).

A. SCREENING RELEVANT STUDIES

The second step after the initial search was the filtration of relevant papers. Duplicate articles found within the database search results were removed. Both exclusion and inclusion criteria were established to ensure the relevance of the articles. Articles that contain attack graphs but inadequately encompassed IoT vulnerability assessment (i.e., make generalizations of the attack graph model having applications towards vulnerability assessment) were excluded. Furthermore, papers focused on areas other than attack graphs, even though they are related to vulnerability assessment, such as using the Petri net model and attack tree models, were excluded. Unpublished papers uploaded to the archive or an extended version of the conference version, as well as papers published in languages other than English, were also excluded. Gray literature, such as predatory journals and conferences, was not seen as a reputable research source. On the other hand, inclusion criteria required that the article must focus on attack graphs, with direct applications to the vulnerability assessment of IoT networks. The article must include empirical data, where data was collected and analyzed, technical evaluations, or case studies of current attack graph techniques for IoT vulnerability assessment.

TABLE 2. The search query for each of the database.

No	Database Name	Search Query	Results
1	IEEE Xplore	("All Metadata ": attack graph) OR ("All Metadata ": vulnerability assessment) OR ("All Metadata ": security weaknesses) AND ("All Metadata ": Internet of things)	3,244
2	ACM Digital Library	[[All: "attack graph"] OR [All: " vulnerability assessment "] OR [[All: " security weaknesses"] AND [All: " internet of things"]]]	412
3	Science Direct	("Vulnerability assessment" OR "attack graph" OR "security weaknesses") AND ("Internet of things")	344

TABLE 3. Applied inclusion and exclusion criteria.

Inclusion Criteria
1. The article must focus on attack graphs, with direct applications to the vulnerability assessment of IoT networks.
2. The article must include empirical data, where data was collected and analyzed, technical evaluations, or case studies of current attack graph techniques for IoT vulnerability assessment. This means that there ought to be some quantified prediction or measurable outcomes comparable with other outcomes from other techniques.
3. The article is included if it contains some model that incorporates partial attack graph applications in IoT networks.
4. The article should be a peer-reviewed journal or conference paper published in English.
Exclusion Criteria
5. Papers focused on areas other than attack graphs, even though they are related to vulnerability assessment, for example, using the Petri net model and attack tree models.
6. Unpublished papers uploaded to the archive or an extended version of the conference version.
7. Papers are published in languages other than English.
8. Gray literature, such as predatory journals and conferences, is not seen as a reputable research source.

This means that there ought to be some quantified prediction or measurable outcomes comparable with other outcomes from other techniques. The article is included if it contains some model that incorporates partial attack graph applications in IoT networks. The article should be a peer-reviewed journal or conference paper published in English.

B. QUALITY ASSESSMENT

The third step involved performing a quality assessment (QA) on the potential papers that had not been excluded in the first and second steps. To ensure that all the primary papers identified contained relevant information for our research area, a methodology for the QA was constructed. The QA applied was based on guidelines proposed by Kitchenham et al. [10] and endorsed by Hosseini et al. [11]. The guidelines were adapted to suit the context of this research area; however, the structure of the questions remains the same. The QA was divided into five stages to systematically check the quality of each included paper (see Table 4). Each of the reviewed articles in the study was assessed using the criteria to ensure they met the quality needed for analysis. Only upon satisfying the entire criterion would a paper be included for analysis.

C. DATA EXTRACTION

Studies faring well in the QA criteria were eligible for data extraction. A data extraction form was created to extract the data systemically and comprised three sections: qualitative data, quantitative data, and contextual data. Once the data extraction form became adequately populated with comparable and relevant quantitative and qualitative data, it was considered suitable for analysis.

The context data includes detailed information such as the type of vulnerability assessment domain the paper focused on, the study objectives, and various other relevant data to the context.

The qualitative data concern the findings of the study and the conclusions made by the authors. Because some papers use qualitative measures to display and record performance, the extracted qualitative data encompassed results that recorded non-numeric values. Some examples of qualitative data include referencing the intuitive nature of the vulnerability assessment, which could be a subjective comment made by the authors or responses by test subjects on the efficacy of the vulnerability assessment solution. The quantitative data encompassed numeric results formulated by measuring the dependent variables. Numeric data that included results sufficiently comparable to other studies were extracted. Some examples include the number of exploits, nodes, the problem size, and the attack graph generation time.

D. DATA ANALYSIS

Numerous data synthesis challenges were faced in this study due to the number of different models, frameworks, or applications used to integrate attack graphs in IoT vulnerability assessment. The challenges were only worsened by the independent variables used to analyze each study's performance. In some cases, quantitative data related to the model's performance indicators had to be contextualized for comparison with other papers. Hosseini et al. [11] suggested it was essential to cross-analyze each model, framework, or application within its context to provide insight. This ensured none of the models, frameworks, or applications were ignored. It was, therefore, necessary to identify any cross-analysis threats between various vulnerability assessment contexts within the research area. Besides collating the qualitative and quantitative data, a meta-analysis was performed to compare the attack graph techniques in their application to IoT vulnerability assessment. A qualitative overview was taken to assess the performance of different models, frameworks, or applications based on their contextual standing within the overall dataset. Some analyzed comparison components include efficiency in finding the shortest path, the algorithm's training period,

TABLE 4. The QA criteria applied.

Stages	QA	QA description
Stage 1	Attack graph construction or application	For a paper to be applicable to the study, it should include the procedure of developing or applying an attack graph concept to an area of vulnerability assessment. Additionally, attack graph models using neural networks, genetic algorithms, machine learning (ML), or artificial intelligence (AI) concepts should undergo a training period that is sustainable and contains varied training data for adequate comparison.
Stage 2	Context	There should be context data supplied within the paper, for example, the specific details of programming languages and algorithms applied to execute the paper's proposed attack graph solution.
Stage 3	Model/ framework details	This was the quintessential criterion for analysis since it involved the assessment of both dependent and independent variables. The dependent and independent variables of the paper were checked to determine if they were established and reported. As well, if the evaluation of an article is quantitative, it should contain both dependent and independent variables to be considered for inclusion.
Stage 4	Data	The paper should outline the data for assessing the model. If testbeds and simulations are used, it is essential to explain the composition of the datasets and how they are standardized for use in an attack graph scenario.
Stage 5	Performance,	The performance of the attack graph model, algorithm, framework, or applications should be measured and accurately presented in the paper.

and performance. While we performed a meta-analysis, none of the models, applications, or frameworks were replicated to confirm the validity of the reviewed studies, as this was beyond the scope of our literature review. Fig. 1 presents the research methodology for the review.

E. RESEARCH QUESTIONS

The process of choosing research questions is a critical step in conducting research. To begin, the researchers must define the research area. In this case, the research area was Attack Graph Analysis for IoT Vulnerability Assessment. Next, the researchers conducted some preliminary reading on the study's subject to gain a deeper understanding of the field and identify the key areas of interest. Based on this preliminary reading, the researchers then identified gaps in the existing literature that needed to be addressed. Finally, the researchers formulated the research questions based on these identified gaps and the information they sought to obtain. The research questions were designed to obtain information on key aspects of Attack Graph Analysis for IoT Vulnerability Assessment, including the parameters used to develop the IoT network attack graph, the model, framework, or application it is based on, the methods used for generating the attack graphs, the effectiveness of the proposed solutions, the tools used for visualizing the model, framework, or application, the properties that can be analyzed, the recommendations for securing the IoT network, and the datasets used to evaluate the proposed solution. Based on these main key aspects, the following research questions guided the literature analysis.

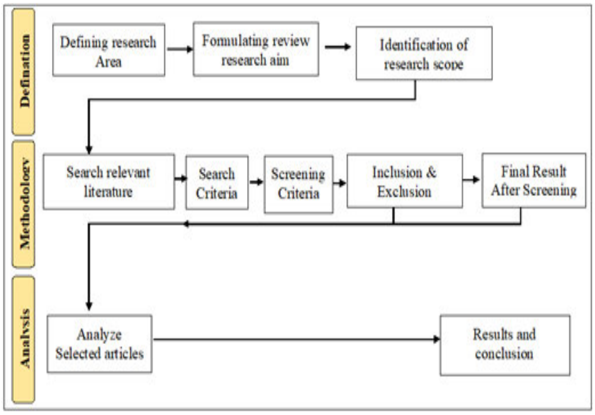
Defination
Methodology
Analysis
Defining research Area
Formulating review research aim
Identification of research scope
Search relevant literature
Search Criteria
Screening Criteria
Inclusion & Exclusion
Final Result After Screening
Analyze Selected articles
Results and conclusion

FIGURE 1. The Literature Search Methodology.

1) What parameters of the IoT network are used to develop the attack graph? [8], [17], [35], [48], [52], [54], [57], [68], [69]
2) What model, framework, or application is the attack graph based on? [43], [50], [56], [67]
3) How are the attack graphs generated? [21], [28], [37], [43], [45]
4) How effectively is the proposed solution managing the attack graph's inputs and outputs? [25], [38] [39], [53], [55], [71]
5) What methods and tools are used for visualizing the model, framework, or application? [6], [16], [27], [30], [37], [39], [40], [62]
6) What properties of the model, framework, or application can be analyzed? [14], [32], [35], [67]
7) What recommendations can be obtained from the attack graph to secure the IoT network? [6], [32], [33], [48], [51]
8) What datasets (empirical, simulated, or hypothetical) were used to evaluate the proposed solution? [7], [20], [21], [33], [39], [54], [67].

F. DEFINITION OF KEY CONCEPTS

Attack Graphs – A succinct representation modeling all possible paths through a network that end in a scenario in which an attacker has successfully achieved their goal [12], [14].

Internet of Things (IoT) – This is a network of physical devices or objects interconnected and equipped with software, sensors, and other technologies to exchange data over the Internet [1], [15].

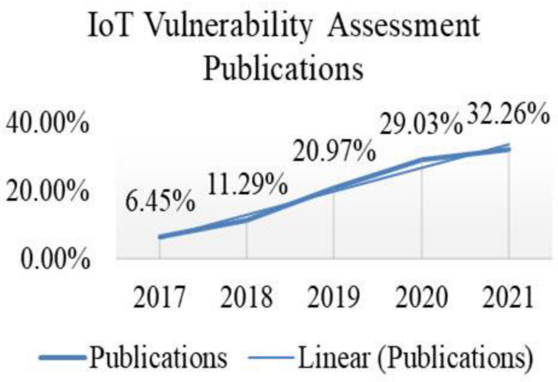

FIGURE 2. IoT Vulnerability Assessment Publications.

IV. RESULT

After searching the five databases using queries specified in Table 2, 4000 papers were found. Most of these papers were duplicated because of the nature of some holistic databases, such as Science Direct and ACM Digital Library, which query other databases. Upon removing duplicated papers, 1442 unique papers remained. After duplicate papers were removed, the exclusion and inclusion criteria specified in Table 3 were applied to the title and abstract of each paper in the remaining dataset. The criterion proceedings brought down the number of papers to 48. Forward and backward snowballing were used to search through the references of the 48 papers to find more papers related to the research area. After the snowballing method, 14 relevant papers were found and thoroughly read using the exclusion and inclusion criteria to determine their relevance to the research area. All 14 papers selected fit the criteria. Consequently, as presented in Table 5, 62 papers were found relevant and included in the literature review analysis.

A. RESEARCH PUBLICATIONS AND DESIGNS

Each article was quantified based on the year of publication to develop an understanding of the trends in this research area. In recent years, there has been a gradual increase in interest in the IoT security vulnerability assessment, as presented in Fig. 2. This is reflected by the number of articles identified: (20.97 %%) in 2019, (29.03%) in 2020, and (32.26%) in 2021. Included in the review, as presented in Fig. 3, 34 (54.84%) are peer-reviewed journal papers and (45.16%) are conference papers. This is an indication of the maturity of the research area. However, it demonstrates that the research area is still very much in its infancy compared to more advanced research topics.

As Fig. 4 presents, there are two major practical motivations for attack graph modeling. The first motivation involves managing the inputs and outputs of the attack graphs effectively. As Shandilya et al. [71] proposed, the first motivation is a methodology challenge, where the inputs (system parameters) must be represented effectively in the model resulting in the attack graph. Furthermore, the model analyses of the various security properties and violation detection should reveal effective responses as the graph outputs. The second motivation is a technology challenge, highlighting the importance of generating the attack graphs autonomously while demonstrating the efficiency and scalability of their applicability in larger systems.

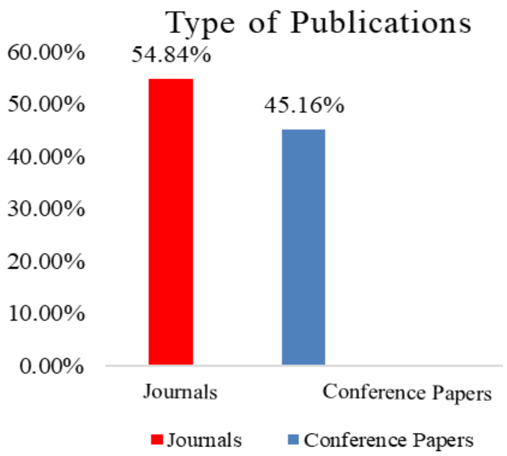

FIGURE 3. The Type of Publication on IoT Vulnerability Assessment.

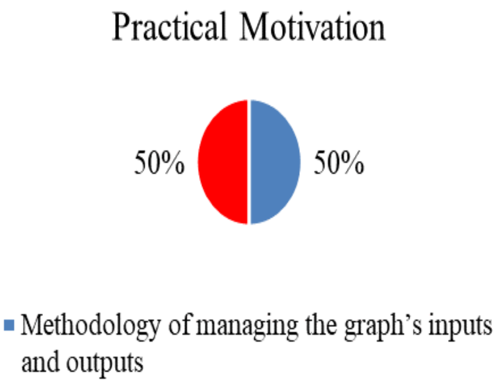

FIGURE 4. The practical motivation of attack graph modeling: about 50% of the authors focus on developing the methodology for managing the graph's inputs and outputs. In comparison, another 50% develop technologies for attack graph generation and analysis.

B. COMPARISON OF ATTACK GRAPH MODELS

The application of modeling within the IoT vulnerability assessment literature manifests itself in several forms. In most articles, many models seem to encompass some level of attack planning, whether through attack graph generation, graph analyses, or other levels of planning. The articles have applied different models, with some studies combining up to six models [27], [46], [57], [69], [70]. This makes it challenging to taxonomize each article based on the general overview of the respective model applied. Most of the studies use some capacity of the MDP, such as POMDP and Bayesian Modeling [34], FPN [67], K-mean clustering [44], or logistical regression models [68], [70]. Difficulties in attaining accurate results using POMDP have been highlighted [34].

TABLE 5. Comparison of designs and contributions.

Ref.	Title	Description	Design	Contribution	Limitation
[16]	A network attack graph generation tool — IEEE CNS 17 poster	Propose a novel method based on core network graphs, to address the complexity of graphs	Simulated Experiment	Propose a tool (Naggen) for generating, visualizing, and analysis of core attack graphs. Demonstrated Naggen's advantages through application in various security applications.	Based on simulation
[17]	Security Modeling and Analysis of Cross-Protocol IoT Devices	Grouped IoT devices according to their communication protocols and developed a graphical security model for devices using a similar communication protocol	Experiment using real networks	The model helped generate sub-networks grouped according to device communication protocols	Uses only one attack model
[18]	A graph-theory-based generic risk assessment framework for the Internet of Things (IoT).	A model-driven risk analysis framework using graph theory	Testbed Experiment	A generic risk assessment framework for IoT systems proposed and implemented based on graph theory	No proof of concept for real IoT systems
[74]	An Experimental Framework for Investigating Security and Privacy of IoT Devices	A framework for investigating privacy and security issues of IoT devices	Testbed Experiment	Can investigate issues of numerous IoT devices such as IP cameras, HDMI sticks, smartwatches, activity trackers, and drones	Based on simulation
[19]	Security Testing Methodology of IoT	A model for testing IoT network integrity to harden the resilience of IoT networks to external attacks	Simulated Experiment	Provides IoT testing methodology	Hardware cracking is irreversible
[20]	Penetration Testing for the Internet of Things and Its Automation	Analyzed IoT security problems and proposed an automated pen testing approach based on BDI modeling	Simulated Experiment	Successful in simulating automated vulnerability assessment using Jason and identifying vulnerabilities in each IoT layer.	Based on simulation
[15]	A Graph-Based Security Framework for Securing Industrial IoT Networks from Vulnerability Exploitations	An IIoT graphical model that focused on addressing the network security issues because of inherent device vulnerabilities	Simulated Experiment	Identification and removal of vulnerabilities with low hop length and high risk.	Based on simulation
[21]	Automatic Generation of Attack Scripts from Attack Graphs	A model that automates network graphs to simulate attack scenarios.	Simulated Experiment	Generates scripts automatically to test different attack scenarios that are generated from an attack graph	Based on simulation
[7]	Attack graph — Based vulnerability assessment of rank property in RPL-6LOWPAN in IoT	An attack graph that investigated the RPL rank property vulnerabilities	Simulated Experiment	A model to prevent the exploitation of the rank property vulnerabilities	Based on simulation
[22]	A Vulnerability Assessment Method in Industrial Internet of Things Based on Attack Graph and Maximum Flow	Using attack graphs and maximum flow to solve the problem of attack path quantification in IoT	Simulated Experiment	Avoiding repetitive calculation and the ability to obtain the probable vulnerability path fast using the augmented road algorithm	Only focused on IIoT application
[23]	Penetration Testing of Intrusion Detection and Prevention System in Low-Performance Embedded IoT Device	A method of hardened security protection by implementing the IDS/ IPS tool Suricata in low-performance IoT devices	Simulated Experiment	A demonstration of a vulnerability assessment approach for Suricata IPS, which could apply to other security rules for embedded IoT devices	Based on simulation
[26]	A New Model for Securing Networks Based on Attack Graph	A novel model using an attack graph to harden network security	Simulated Experiment	Simulation tests and comparisons that demonstrate to possess some reliability and are suitable for small-scale networks	Based on simulation
[13]	Deployment optimization of IoT devices through attack graph analysis	Proposed a model that quantified network security level based on augmented attack graph analysis, accounting for IoT devices' physical location and communication capabilities.	Experimental Case study	The ability of augmented attack graphs in quantifying the security effect of deployed IoT within an organization and the efficiency of optimizing IoT deployment	Deployment simulation for IoT devices

TABLE 5. ***(Continued.)*** **Comparison of designs and contributions.**

[24]	Identification of Critical-Attacks Set in an Attack-Graph	An algorithm that automatically identifies a set of critical attacks, which when blocked result in hardened system security	Experiment using a case study network	Using SCCs of the given attack graph to create an abstracted version	Uses a standard computer and application
[25]	Reality mining and predictive analytics for building smart applications	A system that interacts with wearable medical sensors (temperature sensor, heart rate sensor, and activity sensor) and mobile phones to predict patterns	Experimental Case study	A practical system using real-time gathered data and applicable to any IoT prediction using mobile-generated data and sensors	Findings limited to healthcare sensors
[27]	AVAIN - a Framework for Automated Vulnerability Indication for the IoT in IP-based Networks	Created AVAIN, a model that facilitates automatic scanning of IP-based networks and provides warning of possible vulnerabilities	Testbed Experiment	Provide two simulation scenarios that highlight AVAIN's application in real-world testbeds while using numerous IP-based components.	Does not support Non-IP networks
[28]	Attack graph generation for microservice architecture.	An approach that relates microservices to network nodes	Simulated Experiment	Complete solution, which can be embedded easily in continuous delivery systems.	Based on simulation
[29]	Challenges for Security Assessment of Enterprises in the IoT Era	Three novel ideas that address and overcome using attack graphs for vulnerability assessment	Action research	Use traffic monitoring to leverage passive observations and temporal attack graphs representing the network model at different times	Purely theoretical without implementation
[30]	Analysis of Complex Networks for Security Issues using Attack Graph	A model that identifies the weakest nodes and source of vulnerabilities by depicting the devices as well as data flow	Simulated Experiment	Simplifies the interpretation of attack graphs and reduces the time needed to identify vulnerabilities	Significant time spend filtering false positives
[31]	An Auditing Framework for Vulnerability Analysis of IoT System	An open-source modular approach for auditing IoT device security	Action research	A framework that automates the process of vulnerability assessment and various tools applicable in different segments of the framework	Purely theoretical without implementation
[32]	How Secure Is Your IoT Network?	Evaluated the security and privacy of IoT devices as well as networks	Experimental study	A framework that provides insight into potential attack paths based on their impact, exploitability, or overall risk.	The network flow issues can be inefficient in huge IoT attack circuits
[73]	Security Testbed for Internet-of-Things Devices	A security framework aimed to test all types of IoT devices, considering their different software or hardware parameters.	Testbed Experiment	Demonstrated the testbed operation across multiple IoT devices by employing specific IoT scenarios	Applicability in larger systems is unlikely
[33]	Cost-aware securing of IoT systems using attack graphs	An algorithm that uses compact attack graphs to discover cost-effective vulnerability assessment to harden IoT security	Simulated Experiment	The algorithm computes cost as a function of influence	Based on simulation
[34]	Survey of Attack Graph Analysis Methods from the Perspective of Data and Knowledge Processing	Compared several formal models including the Bayesian model, the Markov model, and the attack graph	Comparative study	Highlights advantages of attack graph over POMDP and MDP	Purely theoretical without implementation
[35]	Testing IoT Security: The Case Study of an IP Camera	Used attack graphs to pentest the vulnerabilities in IP cameras by demonstrating their impact on users' privacy and security	Experimental Case study	Hands-on test on an IP camera focusing on security analysis of the device elements	Based on simulation
[36]	Design and implementation of automated IoT security testbed	Introduced an open-source platform for detecting IoT network weaknesses, focusing on a smart bulb and a wireless camera	Testbed simulation	A system that demonstrates capabilities to examine vulnerabilities in real-world data of two IoT devices: a smart bulb and a wireless camera	Applicability on large-scale networks is unlikely

TABLE 5. *(Continued.)* **Comparison of designs and contributions.**

[37]	A2G2V: Automatic Attack Graph Generation and Visualization and Its Applications to Computer and SCADA Networks	A model for generating and visualizing automated attack graphs	Simulated Experiment	An attack graph is illustrated using computer applications as well as supervisory control and data acquisition (SCADA) networks.	Based on simulation
[38]	Autonomous Security Analysis and Penetration Testing	An autonomous vulnerability assessment and security analysis framework that uses attack graphs to generate a map of security vulnerabilities in the network	Simulated Experiment	Generates automatic vulnerabilities and validates them using practical networks such as enterprise networks	No considerations for unknown vulnerabilities
[39]	Attack Graph Modeling for Implantable Pacemaker	A system, PARMS, for automatically monitoring the pacemaker	Simulated Experiment	Demonstrates the essentiality of configuring the correct security measures in IoT devices such as pacemaker sensors	Based on simulation
[14]	A network attack path prediction method using an attack graph	A model for detecting attack node paths using attack graphs	Simulated Experiment	High accuracy and improved the efficiency in analyzing network security.	Based on simulation
[40]	Automated Vulnerability Testing via Executable Attack Graphs	Automated Vulnerability and Risk Analysis (AVRA), a tool for the identification and exploitation of vulnerabilities, designed for use in vulnerability assessment	Testbed Experiment	A novel approach that enhances the vulnerability assessment process through rigor, repeatability, and objectivity	Based on simulation
[41]	Security Testing Methodology of IoT	A framework designed to detect security vulnerabilities	Testbed Experiment	Demonstrates an automated and adaptable static analysis approach	Based on simulation
[42]	A Lightweight Cyber-Security Defense Framework for Smart Homes	A mechanism for anomaly detection by combining ML and statistical methodologies to detect risks in network traffic time series	Testbed Experiment	Demonstrated high performance of the framework in terms of precision, accuracy, recall, and f-measure	Based on simulation
[43]	Extending Attack Graphs to Represent Cyber-Attacks in Communication Protocols and Modern IT Networks	An extended network security model for MulVAL	Experimental study	Implements a simplified network architecture for both IoT and industrial components	Does not model all network attacks
[44]	Raspberry Pi-Based Intrusion Detection System Using K-Means Clustering Algorithm.	A framework that uses Raspberry Pi to detect, block, categorize, and store the IP address of an intruder	Simulated Experiment	Uses k-means clustering to identify, classify, and block network attacks attempting to breach the host IP	Based on simulation
[6]	IoT-PEN: An E2E Penetration Testing Framework for IoT	An E2E novel vulnerability assessment Framework, IOT-PEN, for IoT devices	Experimental study	A framework that is easily scalable to complex and large IoT networks.	Ineffective in zero-day attacks
[45]	Automatic security management of smart infrastructures using attack graph and risk analysis	A comprehensive automatic technique that uses attack graphs to calculate security indicators, risk assessment, and select protective measures. Uses python	Simulated study	A system for assessing IoT security risks using an automated attack graph	More lead time for security calculation
[46]	A Comprehensive Approach for DDoS Attack Detection in Smart Home Network Using Shortest Path Algorithm	An algorithm that uses input parameters to generate an attack graph to identify the choking node and apply the shortest path to mitigate DDoS attacks	Experimental study	A unique approach that generates an attack graph to detect DDoS using the shortest path algorithm	False positives when exposed to slow-rate attacks
[47]	On the Detection of Persistent Attacks Using Alert Graphs and Event Feature Embeddings	Use feature embeddings obtained from network event logs to construct alert graphs focusing on the host and alert correlation	Experimental study	Generates interpretable attack graphs while extracting causality information that identifies coordinated attacks	Requires manual effort

TABLE 5. ***(Continued.)*** **Comparison of designs and contributions.**

[48]	Research on Automatic Generation and Analysis Technology of Network Attack Graph	Uses dependency attack graph to model edge authority attack graph and introduces the CVSS index of each exploit node to calculate the probability of network risk	Simulated study	The model uses security metrics of a network to scan all vulnerability exploit nodes in the generated graph and calculates as well as sorts vulnerabilities based on their benefits	Purely theoretical without implementation
[49]	An Attack Path Generation Methods Based on Graph Database	Uses a graph database to store host information and the correlation between hosts and the target network's vulnerability information. The graph database query language is used to query and analyze vulnerabilities	Experimental study	A highly scalable graph database, which is superior in handling large volumes of queries and complex correlation logic	Does not support concurrent use by multiple users
[50]	Graph-based Technique for Survivability Assessment and Optimization of IoT Applications	Uses stochastic process-based and graph theory to generate attack graph	Action research	Proposes a model that considers the survivability of IoT system vulnerability assessment in harsh, adversarial, and unfriendly environments	Limited by network topology connectivity
[51]	An Intelligent Recommendation Algorithm for Red Team Strategy in Edge Computing Powered Massive Cyber Defense Exercise	An intelligent recommendation algorithm for launching the proper offence actions corresponding to vulnerabilities	Action research	The model adjusts a given attack graph by adding or removing vulnerabilities as well as measures its difficulty for novice administrators	Purely theoretical without implementation
[52]	Using Deep Learning to Construct Auto Web Penetration Test	Uses CNN to automatically produce the vulnerability assessment code by training the data originating from real attack events	Experimental study	A model based on CNN to automate vulnerability assessment coding using classical attacks, while achieving more execution by conversion of the shell script	Limited to the web-based application
[53]	A Novel Insider Attack and Machine Learning-Based Detection for the Internet of Things	A framework for detecting new insider attacks in IoT effectively using deep learning algorithms to model traffic behavior	Simulated experiment	A low complex machine learning algorithm able to accurately detect insider attacks	Based on simulated data
[54]	Edge-Based Intrusion Detection for IoT Devices	An automated intrusion detection system for IoT devices that uses system-level information (e.g. running process parameters, system calls) to profile devices according to their behavior thus detecting anomalous behavior	Testbed experiment	A system-level model that detects attacks on IoT devices using anomaly detection by constructing baseline behavior profiles	Does not check anomalies in network logs of the IoT devices
[55]	Backdoor Attacks to Graph Neural Networks	Propose a subgraph modeling using the backdoor attack to graph neural networks (GNN) for attack graph analysis.	Experimental study	The GNN model predicts an attacker-chosen target label in the attack graph once they inject a predefined subgraph into the attack graph	Lack of new defences against the backdoor attacks
[56]	Network Security Assessment Based on Full host-based Attack Graph	A framework to construct a full host-based attack graph by splitting the algorithm for weakly connected components	Experimental study	A full host-based attack graph model that evaluates network security from both surface and point perspectives	Limited to virtualization technology
[61]	Security Risk Analysis of Multi-Stage Attacks Based on Data Criticality	Propose security risk metrics to calculate the cost of potential attack based on the criticality of data as well as the dependencies among vulnerabilities	Experimental case study	Metrics based on graphical modeling techniques that consider the combined effects of the criticality of data and vulnerability exploitation on each network asset	The methodology is semi-automatic

TABLE 5. ***(Continued.)*** **Comparison of designs and contributions.**

[63]	Impact Evaluation of DDoS Attacks Using IoT Devices	A hierarchical model that assesses the DDoS effects on system availability of major IT systems and IoT device components	Experimental case study	The model estimated the attack feasibility, attack propensity, pain factor, attacker benefits, and technical ability	Limited to attack-oriented threats from DDoS in IoT devices
[64]	IoT Metrics and Automation for Security Evaluation	Use generic IoT characteristics to propose security metrics and use IoT testbed to develop automation for experimentation with IoT devices	Testbed experiment	Develop novel security metrics for IoT using their security principles and fundamental characteristics that are quantifiable and automated	Focus on generic IoT security principles rather than specific attacks
[65]	Threat Assessment for Power Industrial Control System Based on Descriptive Vulnerability Text	Developed attack graphs using cyber-physical topology and attack sample attributes to evaluate quantitatively the benefits and feasibility of each attack path from vulnerabilities as well as the effect on power IoT devices	Experimental case study	The model analyzes attack objects, attack consequences, and attack methods to establish a textual feature library before constructing attack graphs. The algorithm calculates feasibility and benefits versus the cost of vulnerabilities in power IoT	Limited to power industrial control systems
[66]	GRAVITAS: Graphical Reticulated Attack Vectors for Internet-of-Things Aggregate Security	Propose GRAVITAS, which uses ML to identify undiscovered attack vectors in IoT, while optimizing the placement of defences for cost-effectiveness and optimal performance	Experimental case study	Detect undiscovered exploits using ML facilitating automatic identification of attacks overlooked by manual vulnerability assessment. The exploit scoring system uses the topology vulnerabilities in the attack graph to gauge risk at both the exploit and device levels	The consistency of vulnerability scores among different propagation cycles is difficult to achieve
[68]	Research on Multi-Target Network Security Assessment with Attack Graph Expert System Model	Proposed an optimization algorithm using MulVal attack graph that for acyclic attack graph considers atomic attack weight and attack distance while for simplified attack graph, considers cost versus benefit of security reinforcement	Action research	A methodology for algorithm optimization based on attack path complexity for a generated attack graph after the loop is eliminated. The generated attack graph is optimized to different degrees by calculating attack distance and atomic weight.	No simulated or actual dataset was analyzed to validate the model
[67]	Network Attack Path Selection and Evaluation Based on Q-Learning	Proposed a method using Fuzzy Petri Net (FPN) to establish an attack model, before improving the model using Q-Learning. The attack gain on a power grid was defined from the attacker's perspective to generate the best attack path and analyze the impact on the real-time processes of a power grid	Simulation experiment	An attack model that uses FPN's fuzzy reasoning ability to improve a Q-Learning algorithm then uses the Q-Learning to find the most vulnerable path in the network system	Based on simulation
[69]	Botnet Detection Approach Using Graph-Based Machine Learning	Propose an attack graph model using ML for botnet detection. The model considers the importance of graph features before generating a generalized model to detect botnets using the selected significant features.	Experimental study	An efficient and effective graph-based system for bot detection robust to zero-day attacks and suitable for large-scale systems. The attack graph. The graph-based system for botnet detection is tested on two real datasets.	Limited to network structural features Does not evaluate node attribute features
[70]	Generating Threat Models and Attack Graphs Based on the IEC 61850 System Configuration Description Language	A model using the System Configuration description Language (SCL) to generate attack graphs for vulnerability assessment	Simulated experiment	A methodology for translating SCL to attack graphs to perform attack simulations. The technique allows vulnerability assessment of an electrical substation using configuration files in existence	Based on simulation The generating of the attack graph is not automatic

Others have applied genetic algorithms, which generate solutions via generational fitness iterations [18], [26], [27], [46], [48]. This reflects that these modeling techniques could be the accepted solutions for the attack graph challenges within the research domain.

Furthermore, besides the core modeling techniques, some have applied the fast-forward (FF), contingent fast-forwards (cFF), advanced reinforcement-learning algorithm, and multi-layer hierarchical attack representation models (HARMs) to complement their application [9], [32], [41], [45]. Using such processes to generate attack graphs and planning improves the effectiveness of the modes. Some studies have tried to create shorter attack paths [50], [59], [63], [71] supplemented by contingent options.

C. PERFORMANCE COMPARISON BASED ON VARIABLES

In the evaluated research, several independent variables (IVs) are used to evaluate the proficiency of the proposed frameworks and models. The review determined that while some of the systems examined are analogous, others are not practically comparable. Nonetheless, it is worth highlighting that despite the different approaches, all articles selected have a similar practical objective: furthering the IoT research area on vulnerability assessment. The metrics used by the articles to examine the proficiency of attack graphs within the IoT network domain greatly vary.

The reviewed articles tested their proposed solutions using very unequivocal means. To this end, the independent variables briefly lead to a more identifiable research objective for the desired outcome variable. In contrast, some reviewed studies use common IVs such as network size, algorithm generation, network state, number of exposed hosts, vulnerabilities, number of objectives, the action model, and connectivity. Table 6 presents the summary of the IVs and their represented metrics. Nonetheless, the measurement techniques of the dependent variables (DVs) differ significantly, making it difficult to compare the findings in the same context.

D. IOT SYSTEMS TESTED

Several studies have relatable metrics; nonetheless, the specific IoT network configurations used as IVs for the host's size within a network vary. Fig. 5 depicts the tested IoT system. For example, Agmon et al. [13], Sachidananda et al. [41], and Spanos et al. [42] use the number of hosts as IVs; however, the number of exploits and coverage discovered for each host is the focus of the measurement.

Most of the studies use their own IoT test networks (bespoke networks tailored for the study's needs) to evaluate the performance of the proposed solution. For example, networks range from large-scale and small-scale to industrial IoT networks with devices such as power grids, IP cameras, smart bulbs, wireless cameras, and pacemakers. The critical problem is that such fragmentation makes it challenging to determine the exact difficulty of the IoT test network used in the assessment samples. For that reason, the comparisons between different articles prove difficult.

For instance, what constitutes a small IoT network or an accessible IoT network in one article may not be solved efficiently using another algorithm proposed in another study, thereby making the assessment results applicable only to the proposed model in that article. Surprisingly, none of the studies actively used training as an influential variable within the reviewed articles, despite the strong correlations between algorithm learning and training. Existing techniques for IoT penetration testing can be complex and difficult to understand, leading to inadequate coverage of potential attack scenarios. Additionally, some existing techniques can be time-consuming and resource-intensive, especially when performed manually, and may not be able to adapt to changes in the system or new vulnerabilities that are discovered. Attack graphs address these issues by providing a visual representation of the attack surface and potential attack paths, offering customization to reflect specific vulnerabilities, and allowing automation to streamline the assessment process.

E. PRACTICAL APPLICABILITY OF THE GENERATED ATTACK GRAPHS

Many attempts have been made to achieve the objectives, with various degrees of success. There has been state-space explosion handling while generating the attack graph, graph analyses for security properties and violations, the precision of assessing the attack path efficiently, and making practically implementable recommendations to mitigate the vulnerabilities. As Table 7 highlights, some articles focused on developing technologies associated with attack graph generation and analysis, while others addressed the management of the graph's inputs and outputs. The heat map for the specific focus is presented in Table 7.

Table 7 depicts the distribution of the various practical applicability of the articles on vulnerability assessment within IoT. This heatmap shows the generalization of each practical objective, as naturally, each article has developed its own solutions, which slightly vary in their composition.

V. METHODOLOGY OF MANAGING THE GRAPH'S INPUTS AND OUTPUTS

The first classification of the reviewed studies focuses on managing the attack graph's inputs and outputs. To this end, the contribution of each study based on the IoT system Parameters, metrics, attacking scenarios, and attack models were analyzed. Furthermore, the model analyses for different security properties and vulnerability detection are discussed.

A. FORMAL MODELING OF ATTACK GRAPHS

Zeng et al. [34] compared several formal models and found that obtaining accurate results using the partially observable Markov decision process (POMDP) and Markov Decision Process (MDP) would be challenging because the problem is NP-hard and requires approximation algorithms.

George and Thampi [15] proposed an Industrial IoT (IIoT) graphical model that addressed network security weaknesses because of inherent device vulnerabilities. The proposed IIoT

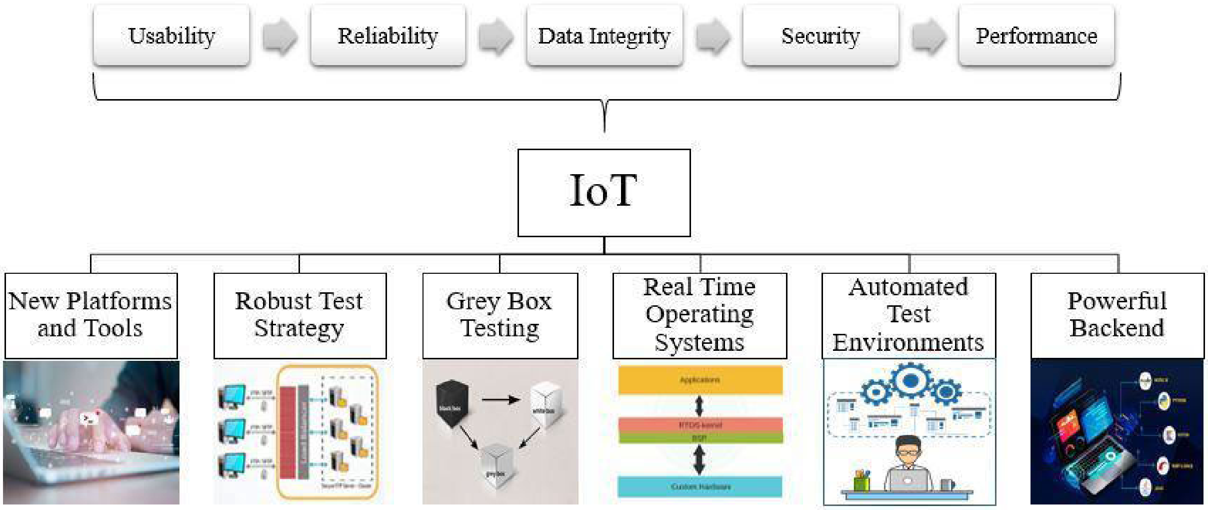

FIGURE 5. IoT System Test Architecture.

TABLE 6. The identified IVs and their metrics.

Independent Variables	Metric	Studies
Network Size	Use standard IV to analyze the performance of their proposed attack graph implementation, where the IV is the size of the network node presented to the attack graph solution	[27, 28, 39, 49, 70]
Number of exposed hosts	The metric is how long it takes for the sample size to be solved or compromised compared to the exploitation steps.	[13, 41, 42, 60, 62]
Genetic algorithms	The metric is the sentient software application to solve the challenges regarding vulnerability assessment by using software evolution to determine the best course of action to compromise the sampled system(s). Most studies achieved this by using the fitness metric to determine how well the solution is suited to the examined process.	[16, 18, 20, 21, 24, 34, 38, 45, 46, 51, 53, 59, 69]
Network state	The metrics used include hosts, vulnerabilities, IoT software, or Network State	[6, 19, 22, 23, 32, 35, 36, 40, 54, 63, 64]
Connectivity	The network connectivity and known weakness are the metrics used to generate the attack graph	[7, 14, 31, 43, 47, 50, 56]
Action Model	The metric is the application of the modeling techniques, where some applied such as FF, cFF, advanced reinforcement-learning algorithm, and HARMs models	[17, 29, 34, 44, 52, 55]
Number of objectives	The metric is the number of exploit objectives required in each vulnerability assessment done, where the measure is the time taken and costs/benefits to solve the weaknesses found (objectives).	[25, 26, 30, 33, 48, 61, 65, 66, 68]
Vulnerabilities	The metric uses vulnerability detection time as a quantifying metric, such as seconds. The measure is required to detect vulnerabilities in the host	[15, 37, 57, 58, 67].

model, which acts as a framework for assessing network risks, includes a set of risk mitigation techniques for network security hardening. The techniques recommended in the study included identifying and removing vulnerabilities with low hop length and high risk.

Egert et al. [27] created AVAIN, a model that facilitates the automatic scanning of Internet Protocol (IP)-based networks and provides warnings of possible vulnerabilities.

The framework facilitates the automated deployment of multiple tools, enabling the development of complex modules for network scanning and analysis. The authors provide two simulation scenarios highlighting AVAIN's application in real-world testbeds using numerous IP-based components.

Tekeoglu and Tosun [74] proposed a framework for investigating IoT security issues. The framework included four constructs, a testbed, topics needing investigation, several experiments for each investigated topic, and a concluding report. The basic technique used in the framework was capturing layer two and layer three packets and then analyzing the packets for numerous features. The framework can investigate networks involving multiple IoT devices, such as IP cameras, HDMI sticks, smartwatches, activity trackers, and drones.

TABLE 7. A summary of models reviewed.

Methodology of managing the graph's inputs and outputs	No
Graph analyses and formulation of security properties	8
system parameters represented in the graph model	7
Formal modeling of attack graphs	8
Vulnerability detection and response formulation	8
Technologies for attack graph generation and analysis	
Automated Graph Generation	10
Graph analysis and violation detection	8
Attack graph response recommendation/ implementation	8
Technology for graph visualization	5

Liu [14] proposed a model for detecting attack node paths using attack graphs. The attack graph was defined during the modeling of the state of vulnerability detected. The network connectivity matrix acquired, the formal vulnerability description, the attack impact, and the obtained attack premise. Consequently, the network path graph was generated to outline the transfer correlation between nodes, map the attack process from one vulnerability or host to the next, and highlight the shortest path to attain the attack intention. Chu and Lisitsa [20] analyzed IoT security problems and proposed an automated vulnerability assessment approach based on belief-desire-intention (BDI) modeling. The vulnerability assessment tools in the perception layer included Hardware Bridge API, Nmap, Openvas, and Nessus. Aircrack-ng was used to check network vulnerabilities, while Fierce and DNSenum collected DNS information for social engineering attacks. Overall, the model simulates automated vulnerability assessment using Jason and identifies vulnerabilities in each IoT layer.

Chowdhury et al. [53] proposed a machine-learning-based framework to detect vulnerabilities of the IoT to insider attacks. The framework uses deep learning algorithms to model traffic behavior and has two components: a gateway (or sink) and a sensor. The algorithms tested, random forest and Extreme Gradient Boosting (XGBoost), showed the ability to accurately detect vulnerabilities to insider attacks with an accuracy of 93%, and Support Vector Machine (SVM) marked 91%. The main contributions include formulating an insider attack that exploits vulnerabilities in the RPL routing protocol.

Stellios et al. [60] developed a model that examines cyber-physical interactions using an attack tree topology. The model uses CVE and CVSS as the building blocks for threat modeling. The main contribution is a model that reduces false positives by classifying identified attack paths based on risk levels, making the methodology efficient in multi-hop attack scenarios. The drawback is that identification of cyber-physical interactions requires manual effort. Brown et al. [66] proposed GRAVITAS, which uses ML to identify undiscovered attack vectors in the IoT while optimizing the placement of defenses for cost-effectiveness and optimal performance. Detecting undiscovered exploits using ML facilitates the automatic identification of attacks overlooked by manual vulnerability assessment. The main contribution is an exploit scoring system that uses the topology vulnerabilities in the attack graph to gauge risk at both the exploit and device levels.

Overall, several models reviewed provide several advantages: low cost, compatibility with available IoT hardware, and open-source software, which makes it possible for practical evaluation, as shown in Table 8.

TABLE 8. A summary of models reviewed.

No.	Ref.	Form of Modeling
1	[74]	● Framework on IoT privacy and security issues
2	[15]	● Checks inbuilt IoT device vulnerabilities ● Use low hop length and high risk
3	[20]	● Automated pen testing BDI modeling ● Tools include Hardware Bridge API, Nmap, Openvas, and Nessus. Aircrack-ng
4	[27]	● Present AVAIN model ● AVAIN scans vulnerabilities in IP-based networks
5	[34]	● Compare the Bayesian model, the Markov model, and the attack graph
6	[14]	● Attack graph to outline the connection between nodes and map the attack process from one vulnerability/ host to the next ● Highlight the shortest attack path
7	[53]	● ML-based framework to detect insider attacks in IoT ● Algorithms using random forest, XGBoost, and SVM models ● Detects insider attack
8	[60]	● CVE and CVSS are the building blocks for threat modeling ● False positives are reduced by classifying identified attack paths based on their risk levels
9	[66]	● The model, GRAVITAS, uses ML to identify undiscovered attack vectors ● Optimizes the placement of defenses for cost-effectiveness and optimal performance

B. GRAPH ANALYSES AND FORMULATION OF SECURITY PROPERTIES

Mathov et al. [29] propose three novel ideas that address and overcome using attack graphs for vulnerability assessment. The authors review the challenges that require solutions when using attack graphs to model and analyze enterprise networks with IoT devices.

The proposed model uses traffic monitoring to leverage passive observations and temporal attack graphs representing the network model at different times.

Agmon et al. [13] proposed a model that quantified network security levels by solving two optimization challenges using the ‘‘depth-first branch and bound (DFBnB) heuristic search algorithm’’: Maximal Utility without Risk Deterioration (MURD) and Full Deployment with Minimal Risk (FDMR). The evaluation made use of the entire network but with IoT device deployment simulation. The model demonstrated the ability of augmented attack graphs to quantify the security effect of deployed IoT within an organization and the efficiency of optimizing IoT deployment.

Sachidananda et al. [41] proposed a framework to detect security vulnerabilities, including Memory Leaks, Buffer Banned functions, Code Injection, and other vulnerabilities. The framework was considered an end-to-end IoT software suite that included protocol stacks, kernels, firmware, Android Packages (APKs), Open-Source Software (OSS), and others.

Sachidananda et al. [41] unpacked and analyzed approximately 21,000 firmware, 50 OSS, and 628 APKs. The framework is an automated and adaptable static analysis approach, which begins with web crawling to fetch the IoT-related files and generates reports that comprise IoT Risk Rating. The framework detected seven new Common Vulnerabilities and Exposures (CVEs) clones in IoT OSS. Over 70% of APKs were vulnerable to Structured Query Language (SQL) Injection, and another 56% used weakly discovered 342 existing

CVEs and 894 susceptible code cryptographic algorithms. The framework also found older versions of BusyBox and 3783 hardcoded passwords in IoT firmware.

Chandan and Khairnar [19] proposed a model for testing IoT network integrity. The testing aims to harden the resilience of IoT networks to external attacks. The model considers four processes. The authors recommended that hardware cracking may be the last resort if phases one to four fail. The devices contain microprocessors and microcontrollers, which store sensitive data that attackers may read.

Nonetheless, hardware cracking is irreversible, and the device may be destroyed, limiting the model's applicability. Skandylas et al. [61] proposed security risk metrics to calculate the cost of potential attacks based on the data's criticality and the dependencies among vulnerabilities. Metrics based on graphical modelling techniques that take into account the effects of how important the data is and how it can be exploited on each network asset are the contribution.

Maciel et al.'s [63] hierarchical model assess the DDoS effects on the system availability of major IT systems and IoT device components. The metrics for the model include attack feasibility, attack propensity, pain factor, attacker benefits, and technical ability. The model is limited to attack-oriented threats from DDoS on IoT devices.

Overall, the reviewed studies provide a theoretical basis for using real-world IoT networks, as Table 9 shows, to deploy vulnerability assessments using attack graphs. This helps the current research quantify the security effect of deployed IoT within an organization and the efficiency of optimizing IoT deployment using attack graphs.

C. SYSTEM PARAMETERS REPRESENTED IN THE GRAPH MODEL

Ge et al. [17] grouped IoT devices according to their communication protocols and developed a graphical security model for devices using a similar communication protocol. Several security models were combined using the cross-protocol devices, and hidden attack paths traversing multiple groups of instruments were computed. The model helped to generate sub-networks grouped according to the device communication protocols, with hierarchical attack representation models (HARM) developed for each sub-network.

The model further generated a meta-HARM that used cross-protocol devices and computed the extended vulnerabilities.

Siboni et al. [73] proposed a security testbed framework to detect vulnerabilities in different IoT systems, considering their different software or hardware parameters. The framework performed standard and advanced network vulnerability assessments. Furthermore, the framework utilized innovative analysis processes using ML algorithms in the testbed to monitor the overall parameters of the IoT device. The framework demonstrated the testbed's operation across multiple IoT devices by employing specific IoT scenarios.

Abdalla and Varol [35] used vulnerability assessment to examine the security weaknesses in IP cameras by demonstrating their impact on users' privacy and network security. The study was conducted using utilities and tools from the Kali Linux platform. The authors performed a hands-on test on an IP camera named "Intelligent Onvif YY HD," focusing on the security analysis of the device elements. The main contribution is a vulnerability assessment technique focusing on security weaknesses.

Hu et al. [48] developed a model for the automatic generation and analysis of attack graphs. The method uses a dependency attack graph to model edge authority in the attack graph. It introduces the CVSS index of each exploit node to calculate the probability of network risk. In this model, the exploit behavior, the initial network conditions, and the attack targets are considered nodes of the attack graph. Due to initial conditions and exploits, the authority is considered directed edges. The main contribution is a model that uses the security metrics of a network to scan all vulnerability exploit nodes in the generated graph.

Jiao et al. [52] used deep learning to construct an automatic model for vulnerability assessment. The model uses CNN to automatically produce a code by training on data from past attack events. The main contribution is a CNN-based model to automate vulnerability assessment using classical attacks while achieving more execution through the conversion of the shell script.

Mudgerikar et al. [54] developed an edge-based vulnerability assessment system for IoT systems. The automated intrusion detection system (IDS) uses system-level information (e.g., running process parameters, system calls) to profile devices according to their behavior, thus detecting anomalous behavior.

Gressl et al. [57] proposed a design space exploration (DSE) framework for checking vulnerabilities in embedded systems, which allows an administrator to specify the system's functionality, model-based attack events, hardware components, and several security functions applicable to the system. The framework extends the classical DSE by incorporating security vulnerabilities using Bayesian Attack graphs. The metrics include general task mapping, calculating security constraints, power consumption, and system performance.

Li and Li [68] proposed an optimization algorithm using the MulVal attack graph, where the algorithm for the acyclic attack graph considers atomic attack weight and attack distance. In contrast, the simplified attack graph considers the cost versus benefit of security reinforcement. Alharbi and Alsubhi [69] developed an attack graph model using ML for botnet detection. The model uses graph feature extraction and normalization to detect botnets using the selected significant features before generating a model. The features include edge degree, edge weight, node centrality, local clustering coefficient, and hub and authority.

Overall, the studies have demonstrated that IoT devices have various system parameters that are quantifiable, as Table 10. demonstrates, making it possible to measure security flaws and weaknesses likely to have multiple

TABLE 9. A summary of models examining graph analysis and their theoretical bas.

No	Ref.	Description	Method
2	[29]	● The proposed model monitors traffic to leverage passive observations and temporal network graphs representing the network model at different times	● Use traffic monitoring to leverage passive observation
3	[13]	● The pentesting model quantifies network security levels based on increased network graph analysis. The graph analysis accounts for the IoT device's physical location and communication capabilities	● Uses DFBnB heuristic search algorithm ● MURD and FDMR ● Accounts for IoT device's physical location and communication capabilities
4	[41]	● The study proposed an automated model that uses an adaptable static analysis approach, commencing with web crawling, and generates reports on the risk rating	● E2E IoT software suite ● Includes protocol stacks, kernels, firmware, APKs, OSS Automated and adaptable static analysis
5	[61]	● Security risk metrics to calculate the cost of potential attack based on graphical modeling techniques	● Analyzes the criticality of data and the dependencies among vulnerabilities
6	[63]	● A hierarchical model that assesses the effects of DDoS on system availability	● Analyzes attack feasibility, attack propensity, pain factor, attacker benefits, and technical ability

TABLE 10. Summary of system parameters represented in attack graphs modeling.

Ref.	Description	Quantified System Parameter
[17]	● The model groups IoT devices based on their communication protocols and then develop a graphical security model for devices that use similar communication protocols.	● Communication protocols ● Transition probability
[8]	● The proposed framework uses innovative ML algorithms to analyze processes and monitor the overall parameters of the IoT device.	● Hardware configurations ● Software configurations
[35]	● The model uses utilities and tools from the Kali Linux platform to perform a vulnerability assessment of privacy and security issues on an IP camera.	● Vulnerabilities on hosts ● Device damage/compromise
[48]	● Uses dependency attack graph to model edge authority in the attack graph and introduces the CVSS index of each exploit node to determine network risks.	● The initial network conditions ● The exploit behavior ● The targets of the attacker
[52]	● Use CNN to automatically produce a vulnerability assessment code based on training data from real attack events	● Classical attacks ● Conversion of the shell script
[54]	● A device-edge split architecture is applied where components running from a server perform the majority of the computational work, while the IoT device components perform minimal work	● Running process parameters ● System calls
[57]	● A DSE framework for testing embedded systems by allowing specification of the system's functionality, model-based attack events, hardware components, and security functions applicable to the system.	● General task mapping ● Security constraints ● Power consumption ● System performance
[68]	● An optimization algorithm using MulVal attack graph for acyclic attack graph, which is based on attack path complexity for the generated attack graph after eliminating the loop	● Atomic attack ● Atomic weight ● Attack distance Cost/ benefit of security reinforcement
[69]	● A technique that considers graph feature extraction and normalization before generating a model to detect botnets	● Edge degree ● Edge Weight ● Node Centrality ● Local clustering coefficient ● Hub and authority

security effects on users. These reviewed works demonstrate the system parameters in the existing literature on attack graph modeling. This informs the current research on the various properties that can be analyzed to help detect a wide range of IoT network weaknesses and vulnerabilities to provide meaningful security recommendations.

D. VULNERABILITY DETECTION AND RESPONSE FORMULATION

Sahay et al. [7] constructed an attack graph investigating the RPL (IPv6 Routing Protocol over Low power and Lossy network) rank property vulnerabilities. All potential threats linked with rank properties were analyzed to construct the attack graph. The results demonstrated that violations of rank property protocols led to several RPL attacks that caused topological isolation, topological sub-optimization, traffic disruption, and resource consumption. The main contribution is presenting a model to prevent the exploitation of the rank property vulnerabilities, while the critical drawback is using a simulated dataset.

Shivraj et al. [18] developed a model-driven risk analysis framework using graph theory. The framework used a bipartite graph approach to envisage risk assessment via attack propagation. An IoT system is modeled in the framework as a DAG with numerous attacks in the form of attack trees and simulation of several attack paths. The framework demonstrates its usefulness with the LINDDUN and STRIDE approaches via empirical experiments and analysis. The study's contribution is a generic risk assessment framework for IoT systems, proposed and implemented based on graph theory. However, there is no proof of concept for real IoT systems.

Cai et al. [26] presented a novel model using an attack graph to harden network security. The model contains networking-fixing techniques for predicting attacks. The authors generate an attack graph based on the minimal cut theorem and shortest path method, allowing the detection of possible attacks, vulnerabilities, and unguarded permissions that need fixing in the network. The main contribution is simulation tests and comparisons that demonstrate some reliability and are suitable for small-scale networks.

Spanos et al. [42] proposed a mechanism for anomaly detection by combining ML and statistical methodologies to detect network traffic time series risks. The framework is a lightweight cybersecurity solution for IoT-based edge computing. The experimental testbed results demonstrated the high performance of the framework in terms of precision, accuracy, recall, and f-measure.

Saxena et al. [46] developed a comprehensive approach to detecting DDoS attacks in the IoT using the shortest path algorithm. The algorithm uses input parameters to generate an attack graph to identify the choking node and apply the fastest path to mitigate DDoS attacks. The parameters used in the algorithm include feature selection, data rate, packet length, and average time. The main contribution is a unique ML approach for generating an attack graph to detect DDoS in the IoT application layer using the shortest path algorithm. However, when exposed to slow-rate attacks, the developed detection engine provided false positives.

Jang et al. [51] developed a recommendation algorithm for the red team strategy. The intelligent recommendation algorithm recommends the proper offensive actions corresponding to detected vulnerabilities. The main contribution is a model that adjusts a given attack graph by adding or removing vulnerabilities while measuring their complexity for novice system administrators.

Setzler and Mountrouidou [64] use generic IoT characteristics to propose IoT vulnerability assessment metrics. The study uses an IoT testbed to develop automation for experimentation with IoT devices. The contribution is the development of novel security metrics for the IoT using their security principles and fundamental characteristics that are quantifiable and automated.

The main contribution of the reviewed studies on the use of attack graphs for violation detection is a basis for developing a framework that considers resource availability and understandability by non-security experts. Table 11 summarizes the literature focus, and Fig. 6 presents the vulnerability detection and response formulation.

VI. TECHNOLOGIES FOR ATTACK GRAPH GENERATION AND ANALYSIS

The second classification of the reviewed studies focuses on the technology challenge, which highlights the importance of generating the attack graphs autonomously while demonstrating the efficiency and scalability of their applicability in vulnerability assessment in larger systems. The attack graph generation, analysis, and visualization processes are presented in Fig. 7.

TABLE 11. Vulnerabilities detected in attack graphs modeling.

Ref.	Quantified Violation Detection
[18]	• Predict risk via attack propagation.
[7]	• Detects violation of rank property protocols
[26]	• Provides networking fixing techniques by predicting attacks
[42]	• Uses ML to detect risks in network traffic time series
[46]	• Quantifies feature selection, data rate, packet length, and average time
[51]	• Adjusts a given attack graph by adding or removing vulnerabilities
[64]	• Novel IoT security metrics using device security principles and fundamental characteristics that are quantifiable as well as automated

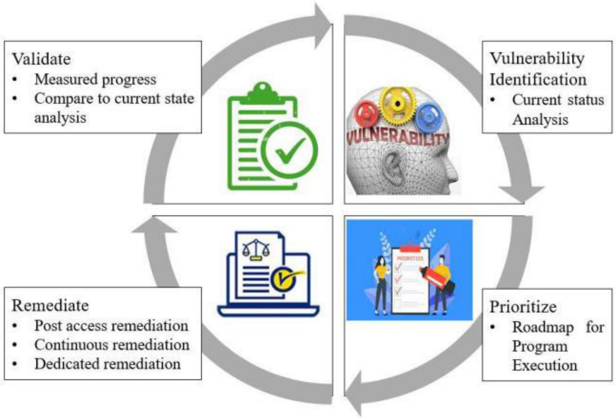
Validate
• Measured progress
• Compare to current state analysis
Vulnerability Identification
• Current status Analysis
VULNERABILITY
Remediate
• Post access remediation
• Continuous remediation
• Dedicated remediation
Prioritize
• Roadmap for Program Execution

FIGURE 6. The Vulnerability Detection and Response Formulation.

A. AUTOMATED GRAPH GENERATION

Yadav et al. [6] introduced a novel framework, IOT-PEN, for IoT devices. The IoT-PEN follows a client/server architecture where ‘‘a system with resources’’ acts as a server and IoT nodes as clients. The framework is a scalable, end-to-end, automatic framework for detecting different vulnerabilities that can be breached on the targeted system using attack graphs. The study recommends prioritization by identifying critical paths for efficient patching. The main contribution is a framework that can be easily scaled to complex and large IoT networks.

Stan et al. [43] presented an extended network security model for Multi-host, Multi-stage Vulnerability Analysis Language (MulVAL). The model considers the topology of the physical network, supports short-range wireless IoT network protocols, models attacks in the design stage of network protocols, and models specific industrial networking architectures. Numerous attack models were studied for man-in-the-middle, spoofing, and denial-of-service (DOS) attacks.

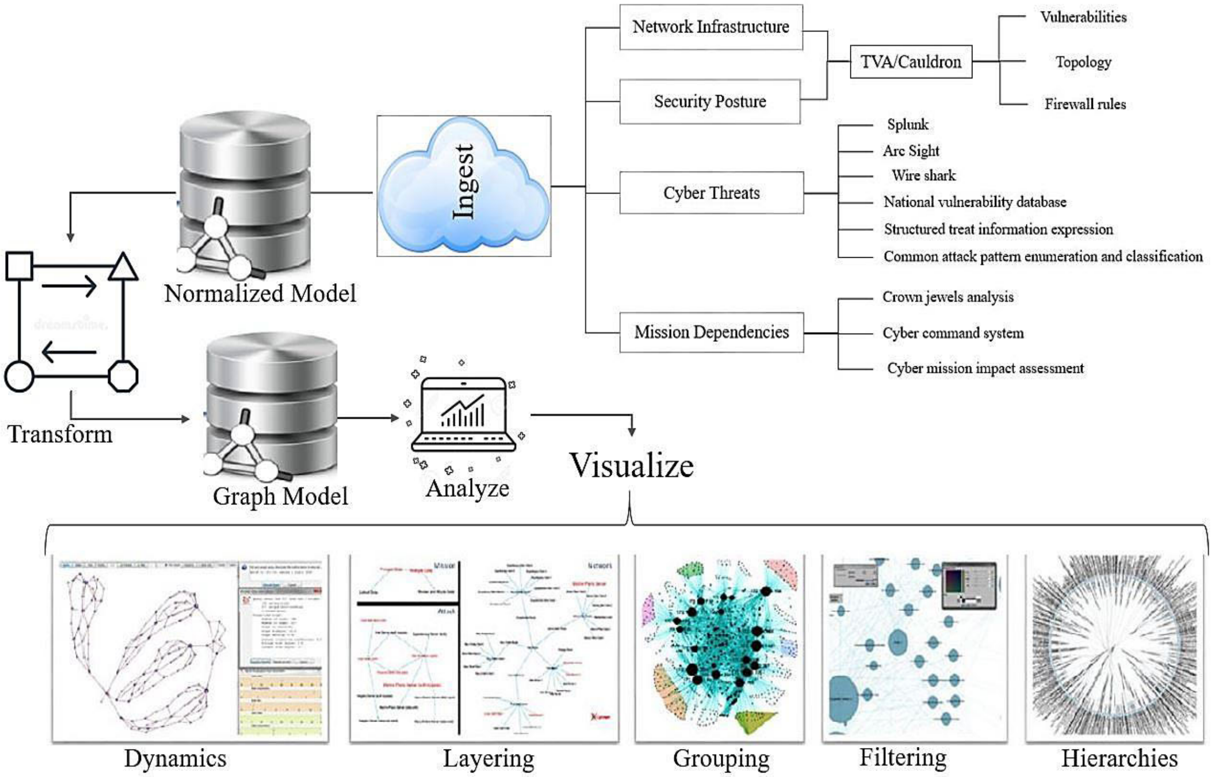

FIGURE 7. Attack Graph Generation, Analysis, and Visualization Process; adapted from [68].

The key contribution is a model that implements a simplified network architecture for IoT and industrial components.

Payne et al. [32] evaluated the security and privacy of IoT devices and networks. The study demonstrated the efficacy of attack circuits as reliable tools for computing security standards. The authors propose a framework for creating attack circuits using natural language processing (NLP) to construct input/output pairs. Standard security scoring measures are also used to compute the weights, and efficient optimization techniques are applied to evaluate attack circuits. The contribution is a framework that provides insight into potential attack paths based on their impact, exploitability, or overall risk.

Nichols et al. [21] presented a model that automates attack graphs to simulate attack scenarios. The model is a procedure for automatically generating scripts to test different vulnerabilities generated from an attack graph. An intermediary program is then used to execute a simulation of the scenarios.

Al-Ghazo et al. [37] proposed a model for generating and visualizing automated attack graphs. The model's algorithm uses existing tools and utilities to generate an attack graph that enumerates all possible system vulnerabilities that may be exploited. A formal network representation is captured using the architecture description tool, their pre-test and post-test conditions, and specific security properties. The contribution is an attack graph illustrated using computer applications, supervisory control, and data acquisition (SCADA) networks.

Ibrahim et al. [28] proposed an approach that related microservices to network nodes. The approach generates attack graphs that can be used to discover, analyze, and mitigate possible attack paths in their container and microservice-based networks. Their contribution is a complete solution that can be embedded easily in continuous delivery systems. The study demonstrates the scalability and efficiency of the approach based on a simulated real-world scenario, which is also a drawback of the study.

Ivanov et al. [45] proposed automatic security management for the IoT using risk analysis and attack graphs. The comprehensive automated technique uses Python to generate attack graphs that calculate security indicators, risk assessments, and special protective measures. Yuan et al. [49] proposed a method for attack graph generation using a graph database. The method creates a Neo4j graph database to store host information and the correlation between hosts and the target network's vulnerability information.

Shakhov and Koo [50] combine stochastic process-based models and graph theory to generate attack graphs. The model considers the survivability of IoT systems by checking vulnerabilities in harsh, adversarial, and unfriendly environments. The authors provide a quantitative method to assess IoT system survivability by combining the specificity of intrusion details, network topology, and properties of intrusion prevention and detection systems. The model developed is Markov chain-based and characterizes individual system availability.

Wang et al. [56] use an entire host-based attack graph framework to assess IoT network security. The framework constructs a full host-based attack graph by splitting the algorithm into weakly connected components. The attack graph is generated using network information, including topology, node, and vulnerability information.

Wu et al. [67] used Fuzzy Petri Net (FPN) to establish an attack model before improving the model using the Q-Learning algorithm. The attack gain on a power grid was defined from the attacker's perspective to generate the best attack path and analyze the impact on the real-time processes of a power grid. The contribution is an attack model that uses FPN's fuzzy reasoning ability to improve a Q-Learning algorithm, and then uses the Q-Learning algorithm to find the most vulnerable path in the network system. The evaluation and validation of the model based on a simulated database is the main drawback. Table 12 presents the summary of the reviewed models based on automatic graph generation.

The attack graph, as shown in Figure 8, can provide a clear understanding of the security status of an IoT network. In one scenario, all vulnerabilities are patched and secured, indicated by green nodes with the ''Patched'' status. In another scenario, the network has unpatched vulnerabilities, indicated by blue nodes with the ''Vulnerable'' status. The attack starting point is the same in both scenarios, but in the scenario with unpatched vulnerabilities, the attacker is able to exploit these vulnerabilities and access the user host or intranet database server, as indicated by the edge label ''Exploits''. This demonstrates the importance of regularly patching vulnerabilities to prevent successful attacks.

It is also important to monitor and analyze the attack graph, as shown in Figure 8, to identify and secure any hidden paths. In the scenario with all vulnerabilities patched, the attacker is unable to exploit vulnerabilities and no hidden paths are discovered. However, in the scenario with unpatched vulnerabilities, the attacker is able to exploit these vulnerabilities and potentially discover hidden paths. By understanding the attack graph paths and parameters, organizations can better defend their IoT networks against potential threats and improve overall security.

In the experimental studies identified, there were two vulnerabilities in the H1 Web server in the IoT context. The Apace software flaw is primarily responsible for the vulnerability. On the H2 MySQL database server, most database system vulnerabilities will affect the storage data. Security; H4 user host system is Windows XP. As the common user operating system has many vulnerabilities, attackers likely to use it as a springboard for an attack on other servers. The H3 FTP server host system is a Windows system with many vulnerabilities, some of which are vulnerabilities in the FTP service software itself. The attacker has normal access to the internal network's web server and FTP server. The firewall is unable to identify the attacker's attack strategy. The attacker's host H, serves as the attack's starting point. As shown in the figure below, forward and backward searches are combined to create an attack graph in the IoT environment during the attack graph generation process. The attack graph generation process will coincide with the vulnerability attack's climax. As the vulnerability shifts, the redundant nodes in the diagram will undergo a process of reproduction and evolution under the control of the intelligent early warning algorithm. To access the user host or the intranet database server, the attacker first uses the H-based platform to exploit vulnerabilities in the remote Web server and FTP server for penetration and privilege escalation.

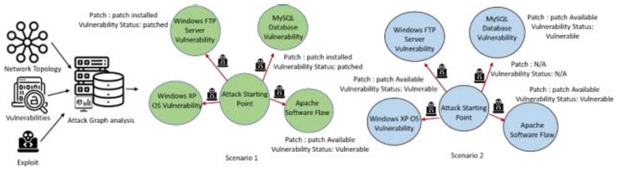
Network Topology
Vulnerabilities
Exploit
Attack Graph analysis
Patch : patch installed
Vulnerability Status: patched
Windows FTP Server Vulnerability
MySQL Database Vulnerability
Patch : patch installed
Vulnerability Status: patched
Windows XP OS Vulnerability
Attack Starting Point
Apache Software Flaw
Patch : patch Available
Vulnerability Status: Vulnerable
Scenario 1
Windows FTP Server Vulnerability
MySQL Database Vulnerability
Patch : patch Available
Vulnerability Status: Vulnerable
Patch : N/A
Vulnerability Status: N/A
Patch : patch Available
Vulnerability Status: Vulnerable
Patch : patch Available
Vulnerability Status: Vulnerable
Windows XP OS Vulnerability
Attack Starting Point
Apache Software Flaw
Scenario 2

FIGURE 8. Attack Graph Comparison of Patched (Scenario 1) and Unpatched (Scenario 2) IoT Networks.

At the end of this section, it is evident that effective generation techniques are crucial for attack graph research. The generation methods presented in these studies lay the foundation for further analysis, visualization, and implementation of attack graphs. In the following section, various approaches for using these generated attack graphs for violation detection and vulnerability assessment will be discussed.

B. GRAPH ANALYSES AND VIOLATION DETECTION

Wang et al. [22] used attack graphs and maximum flow to solve path quantification attacks in industrial IoT. The technique considers the correlation of network nodes and factors influencing the attack behavior. Attack risk is computed using CVSS, which improves the degree of attack path quantification. The method's contribution is avoiding repetitive calculation and quickly obtaining the possible vulnerability path using the augmented road algorithm. The study results demonstrate the approach is feasible and can objectively evaluate vulnerability and risk path.

Malzahn et al. [40] proposed Automated Vulnerability and Risk Analysis (AVRA), a tool for identifying and exploiting vulnerabilities designed for vulnerability assessment. The tool's advantage is that it assesses an entire network and integrates network and host information to create an attack graph. AVRA was tested successfully in a virtual environment to demonstrate usability and practicality. The tool's contribution is a novel approach that enhances vulnerability assessment through rigor, repeatability, and objectivity.

Al-Ghazo and Kumar [24] presented an algorithm that automatically identifies a set of critical vulnerabilities, which, when blocked, result in hardened system security. The authors utilize the Strongly-Connected-Components (SCCs) of the given attack graph to create an abstract version. The validation and implementation of the algorithm occur in real-world settings using a case study and a SCADA network for a water treatment plant. Chowdary et al. [38] proposed an autonomous vulnerability assessment framework that uses attack graphs to generate a map of security vulnerabilities

TABLE 12. Violation detected in attack graphs modeling.

Ref.	Description	Generation Method
[21]	• Present a model that automates attack graphs by generating automatic scripts to test different attack scenarios	• Generates automatic scripts • Mapping assets • Simulation using the simManager program
[32]	• A framework for creating attack circuits using NLP to construct input/output pairs.	• Uses NLP to construct input/output pairs • Standard security scoring measures used to compute the Attack weights
[43]	• The study presents a network security model MulVAL, which models specific industrial networking architectures	• Model for MulVAL • Considers the topology of the physical network
[28]	• A model that can discover, analyze, and mitigate possible attack paths in their container and microservice-based network	• Relates microservices to network nodes • Discover attack paths in their container and microservice-based networks
[6]	• A vulnerability assessment framework for IoT devices, which follows a client/server architecture where "a system with resources" acts as a server and IoT nodes as clients	• Follows a client/server architecture • Scalable, E2E, automatic,
[37]	• A model for generating and visualizing automated attack graphs using existing tools and utilities	• Uses computer applications and SCADA networks
[45]	• Automatic security management for IoT using risk analysis and attack graph	• Uses Python, graph description language (DOT)
[49]	• Develop a Neo4j graph database to store host information and the correlation between hosts and the target network's vulnerability information. An algorithm queries the database for vulnerability information such as attack paths, and network topology, before presenting a visual display to a user	• Defines a method to store data on network vulnerability on a graph database • Uses graph database query language • The algorithm takes the target network's graph data as input and outputs all possible attack paths The algorithm can query the highly scalable graph database
[50]	• Generate an attack graph by combining stochastic process-based models and graph theory	• A quantitative method to assess IoT system survivability • Use intrusion details, network topology, and intrusion prevention/ detection system properties
[56]	• Generate an attack graph using network information, i.e., topology, node, and vulnerability information	• A full host-based attack graph • Dijkstra algorithm
[67]	• An attack model for vulnerability assessment is a power grid, defined from the attacker's perspective to generate the best attack path and analyze the impact on the real-time processes	• Fuzzy Petri Net (FPN) • Q-Learning algorithm

in the network. The framework uses a learning algorithm based on Deep-Q Network (DQN) to pinpoint the optimal policy for vulnerability assessment. The framework generates automatic vulnerabilities and validates them using practical networks, such as enterprise networks.

Musa et al.'s [30] study suggested a model that identifies the weakest nodes and sources of vulnerabilities by depicting the devices and data flow. The model reduces the complexity of attack graphs using MulVal and CVSS base scores as the assessment criteria. Burr et al. [47] proposed using event feature embeddings and alert graphs to detect persistent vulnerabilities. The feature embeddings are obtained from network event logs to construct alert graphs

Focusing on the correlation between the host and the alert. The constructed graph involved IP nodes and alert nodes as well as internal IP edges and alert edges.

Zhang et al. [55] propose subgraph modeling using the backdoor attack to neural graph networks (GNN) for attack graph analysis. The GNN model predicts an attacker-chosen target label in the attack graph once a predefined subgraph is injected into the attack graph. The main contribution is a GNN classifier, trained using the backdoored training dataset, to accurately predict the target label for the attack graph once the same subgraph is injected.

Liu and Zhao [59] developed an algorithm to calculate attack paths and discover vulnerable nodes. Distribution electronic stations are used to construct a simulation program for power IoT attacks. The attack simulation tool uses an attack graph to test an electronic distribution station's security status component, vulnerability values, and optimal attack path.

Overall, the reviewed studies use various methods for generating the attack graph, such as MulVal, as Table 13 shows. The feasibility of several methods for evaluating vulnerability and risk path was provided. The networks tested range from small networks to enterprise networks to large-scale SCADA networks. The interpretation of attack graphs was simplified, thus reducing the time needed to detect vulnerabilities and their origin in the network.

In summary, violation detection techniques play a significant role in identifying vulnerabilities and risks in IoT networks. These methods utilize the generated attack graphs from section A to assess network security. In the next section, various approaches for using visualization techniques to aid in understanding and interpreting these attack graphs will be explored, thereby enhancing violation detection and vulnerability assessment.

C. VISUALIZATION OF THE ATTACK GRAPH

Asri et al. [25] developed a model that interacts with wearable medical sensors (temperature sensor, heart rate sensor, and activity sensor) and mobile phones to predict vulnerabilities.

TABLE 13. A summary of models focusing on graph analysis and violation detection.

Ref.	Description	Analysis Method
[22]	• Uses attack graphs and maximum flow to quantify attack path in IoT	• Computer attack risk using CVSS • Uses an augmented road algorithm
[24]	• An algorithm that uses network SCCs to automatically identify a set of critical attacks which, when blocked, result in hardened system security	• Use SCCs • Experiment with the SCADA network
[30]	• A model that reduces the complexity of attack graphs using MulVal and CVSS	• Identifies the weakest nodes • Uses MulVal and CVSS base scores as the assessment criteria
[38]	• A vulnerability assessment framework using an algorithm based on DQN to generate a map of security vulnerabilities in the network	• Learning algorithm based on DQN
[40]	• The study presents AVRA, a vulnerability assessment tool for the identification of vulnerabilities	• Assess an entire network • Integrates both network and host information
[47]	• A technique using event feature embeddings and alert graphs to detect persistent attacks.	• The algorithm learns embedding from network event logs • Compares IP nodes and alert nodes as well as internal IP edges and alert edges
[55]	• A subgraph modeling using the backdoor attack to GNN for anomaly violation and analysis.	• A GNN classifier trained using a backdoored training dataset • Accurately predicts the target label for the attack graph once the same subgraph is injected into it
[59]	• An algorithm calculates attack paths and discovers vulnerable nodes using electronic distribution stations to construct a simulation program for power IoT devices.	• Quantifies difficulty of exploiting security vulnerabilities • The severity of the consequences after exploiting security vulnerabilities

The strategies have used an Arduino for data collection from the health sensors, a Raspberry Pi 3 for processing and programming, and the K Means clustering algorithm for pattern prediction. The system uses real-world data managed and processed over Apache Spark Databricks. The main contribution is a practical system based on real-time data collection that can be applied to any IoT prediction based on mobile-generated data and sensors. The findings are, however, limited to healthcare sensors.

Ibrahim et al. [39] proposed PARMS to monitor pacemakers' vulnerabilities automatically. The system uses architecture analysis and design language (AADL), checked using the JKind model checker. The Graphviz tool is used to visualize the generated attack graph and categorizes security attacks based on the violation of the security features. The contribution is an attack graph that demonstrates the essentiality of configuring the proper security measures in IoT devices such as pacemaker sensors. Nevertheless, the evaluation is based on a simulated dataset.

Barrère and Lupu [16] described the complexity of attack graphs as the key challenge in the practical application of security assessment. The complexity arises as networks become denser and larger, which inherently defies the graph's scalability aspects at both the computational level and from the perspective of human understanding. Consequently, applying attack graphs to dense scenarios can yield cyclic and complex attack graphs. The authors propose a novel method based on core attack graphs to address the scalability of charts. Finally, the study proposes a tool, Naggen, to generate, visualize, and analyze core attack graphs. The significant contribution is demonstrating Naggen's advantages through application in various security applications. The major drawback is the evaluation of Naggen using a simulated dataset.

Yiğit et al. [33] suggested an algorithm that uses compact attack graphs to discover cost-effective vulnerability assessments to recommend IoT security measures. At first, all likely attack paths are first identified, and then initial or exploit conditions are used with minimum effective removal. The algorithm computes cost as an influence on vulnerabilities and removal costs. The process iteratively continues until the total cost exceeds the allocated budget. Nevertheless, validating the algorithm using a simulated experiment is a significant limitation.

Muhati and Rawat [58] apply the ML-based hidden Markov model (HMM) to predict the agility of vulnerability assessment in the IoT. The technique uses HMM for prediction, projection, and cyber-visualization to facilitate precise vulnerability assessment. Muhati and Rawat [58] developed a prototype as a web service, which queries the data required from pre-defined text files and virtual nodes setup and sends it as HMM output to a front-end visual display module. The module is written in JavaScript and C# and is published through the new Unity software Entity Component System (ECS). The major drawback is that IoT vulnerabilities must remain constant (only changing based on prior successful attacks).

Liu et al. [62] developed a game attack–defense graph (GADG) technique that incorporates the attack graph and

TABLE 14. **A summary of models focusing on graph visualization.**

Ref.	Description	Visualization of Graph
[16]	• A vulnerability assessment tool, Naggen, for generating, visualizing, and analysis of core attack graphs	• Based on core attack graphs Propose a technique for graph visualization
[25]	• A model that interacts with wearable medical sensors and mobile phones to predict attack patterns	• Use Arduino for data collection • Raspberry Pi 3 for processing and programming • K Means clustering algorithm for pattern prediction • Processed over Apache Spark Databricks
[33]	• An algorithm that uses compact attack graphs to discover cost-effective vulnerability assessment for IoT devices	• The algorithm computes cost as a function of influence
[39]	• Presents PARMS for automatically monitoring the pacemaker	• Uses AADL and JKind model checker A. A Graphviz tool used for visualization
[58]	• Apply ML-based HMM to predict the agility of intrusion detection in IoT and visualizes the output using a prototype as a web service	• Pre-defined text files and virtual nodes setup sent to a front-end visual display module • Visual module developed using JavaScript and C# programming languages • Published from new Unity software ECS
[62]	• A game attack–defense graph (GADG) technique that incorporates the attack graph and the game theory to model vulnerability assessment in a local metering system (LMS)	• GADG technique • Mixed-strategy Nash equilibrium

the game theory to model vulnerability assessment in a local metering system (LMS). The authors use an attack graph to visualize different LMS cyberattack paths and their effects.

The main contribution is generating an attack graph using game theory models to attain optimal vulnerability assessment for the attack scenario. The technique is limited to LMS applications. Overall, the main contribution of the reviewed studies on the visualization of attack graphs, as shown in Table 14, is the demonstration of using algorithms that scale linearly with network size and hence could enable the applicability of attack graphs to large networks with a large number of IoT nodes.

Visualization techniques discussed in this section are essential for human understanding and interpretation of the attack graphs generated in section A and utilized in section B for violation detection. Clear and effective visualization can help security analysts make better-informed decisions regarding vulnerability assessment and remediation. In the following section, the practical implementation of attack graphs will be explored, which can benefit from the advancements made in generation, violation detection, and visualization techniques.

D. ATTACK GRAPH IMPLEMENTATION

Zitta et al. [23] proposed a method of recommending security protection by implementing the Intrusion Detection System (IDS) or Intrusion Prevention System (IPS) tool Suricat in low-performance IoT networks. The study further proposes vulnerability assessment focusing on software tools, including Metasploit and NMAP. The IDS/IPS tool Suricata is integrated into the Raspberry Pi 3. The main contribution is demonstrating a vulnerability assessment approach for Suricata IPS, which could apply to other security rules for embedded IoT devices. Nonetheless, using simulated data to evaluate Suricata is the key limiting factor.

Sumanth and Bhanu [44] presented a framework that uses Raspberry Pi to detect the IP address of an intruder. The framework implements a vulnerability detection scheme using the k-means clustering algorithm approach. The detection framework scheme focuses on IP signatures. The key contribution is a k-means clustering framework to identify, classify, and block network attacks attempting to breach the host IP. The major drawback is the validation of the framework using simulated data.

Nadir et al. [31] proposed an open-source modular approach for auditing IoT device security. The framework covers firmware, hardware, and communication vulnerabilities. The modular approach uses existing open-source utilities and tools to implement the proposed framework. The standout parameters in the proposed framework include modular design, scalability, extensibility, and assessment. The authors further highlight various tools applicable to different segments of the framework. The validity and feasibility of the framework are tested by the vulnerability assessment of an IoT network; however, the lack of implementation is a shortcoming.

He et al. [65] developed attack graphs using cyber-physical topology and attack sample attributes to evaluate each attack path's benefits and feasibility. The attack graph assesses vulnerabilities and their effect on power IoT devices. The contribution is a model that analyzes attack objects, attack consequences, and attack methods to establish a textual feature library before constructing attack graphs. The algorithm calculates the feasibility and benefits versus the cost of

TABLE 15. A summary of models on graph implementation.

Ref.	Description	Graph Implementation
[23]	• A method of implementing vulnerability assessment in low-performance IoT devices	• Suricata, IDS/IPS tool is integrated into Raspberry Pi 3
[31]	• An open-source modular approach for auditing IoT device security	• Open-source modular approach • Covers firmware, hardware, and communication vulnerabilities • Modular design, scalable, extensible, automatic
[44]	• A model that implements a detection scheme using the k-means clustering algorithm	• Uses Raspberry Pi to detect, block, categorize, and store the IP address of an intruder
[65]	• Uses cyber-physical topology and attack sample attributes to evaluate the benefits and feasibility of each attack path quantitatively	• Analyzes attack objects, attack consequences, and attack methods to establish a textual feature library
[70]	• Use System Configuration Description Language (SCL) to generate attack graphs for vulnerability assessment in electrical substations	• Analyzes existing configuration files

vulnerabilities in power IoT implementation. Nevertheless, the model was tested only on industrial power control systems.

Rencelj Ling and Ekstedt [70] used the System Configuration Description Language (SCL) to generate attack graphs for vulnerability assessment. The study provides a methodology for translating SCL to attack graphs to perform attack simulations. The main contribution is a technique for vulnerability assessment in electrical substations using configuration files. The main drawback is that generating the attack graph is not automatic. Overall, the studies have demonstrated that vulnerability assessment using attack graphs could be applied to specifying security rules for embedded IoT systems. The major contribution of the reviewed studies is the steps for implementing a framework that automates the vulnerability assessment process using attack graphs, as shown in Table 15.

As a result, practical implementation of attack graphs is essential for ensuring that the research findings and advancements made in generation, violation detection, and visualization techniques are effectively applied in real-world scenarios. The studies discussed in this section showcase various applications of attack graphs in IoT networks, highlighting their significance in vulnerability assessment and remediation. By leveraging the knowledge and techniques from sections A, B, and C, practical implementations of attack graphs can lead to enhanced network security and better protection of IoT devices against cyber threats.

VII. DISCUSSION AND FUTURE DIRECTIONS

The comprehensive critical examination of key works in sections III-V presents the state of the art in attack graph modeling and application in IoT networks. The ineffective usage of state-space explosion in attack graphs has been hampered by its complexity and the fragmentation of IoT systems. Recent work has seen an increase in the number of system parameters represented in attack graph modeling, resulting in a wide variety of parameters being studied, allowing for the identification of multiple security. Weaknesses and providing practical security recommendations.

Deep learning technology has matured to the point where it can learn progressively from IoT networks and generate attack graphs with 1000+ nodes dynamically. The generation of attack graphs and their representation can be replicated effectively using tools like MulVal, as demonstrated in [34], [47], [72]. Once the methodology has been determined, the attack graphs can be automatically generated in a scalable manner to include up to 1000+ nodes. Furthermore, as the process becomes increasingly automated, identifying attack pathways and subgraphs of interest becomes more a function of machine functionality than human effort. Effective approaches have been devised to handle or circumvent the increasing state-space complexity.

The network's linear temporal logic describes the security properties reviewed. The attack pathways will be generated if any of these properties are violated. Various approaches are used to detect property violations on both the entire network and individual hosts. Ranking the nodes, separating the topological and vulnerability information, and other methods are used to visualize the graphs. To this end, the methodologies include separating network-topology and host-vulnerability information in the graph, ranking the nodes in the graph, and creating a subgraph. Visualization tools that represent and visualize the graph include Apache Spark Databricks, Graphviz, and NetSpa. The majority of the recommendations are based on static analysis. To that end, the budget influences the steps that would aid in preserving the higher-priority security properties. Using attack graphs to develop a metric for identifying zero-day attacks remains a significant challenge. This is because it is crucial to evaluate the weaknesses associated with the possible exploitation of unknown vulnerabilities and the hardening of the security of the IoT network related to such vulnerabilities. Several attempts have been made to achieve attack graph scalability in the IoT literature.

Nonetheless, striking a practical balance between effectiveness and scalability remains a key knowledge gap identified, especially as IoT systems become more complex. For example, several machine learning (ML) models

TABLE 16. State-of-the-art attack graph for IOT vulnerability assessment.

Features	Prominent Works
Methods used/ Formal models	Bayesian Model, Markov Model, BDI, Heuristic Search algorithm, HARM, Natural Language Processing (NLP), Omega Languages, Raspberry Pi Linear Temporal Logic, Python, Graph Database Query Language, Stochastic Process, CNN, Fuzzy Petri Net, MulVal, System Configuration description Language (SCL), Architecture Analysis and Design Language (AADL)
Parameters represented	Network size, number of exposed hosts, genetic algorithms, network state, connectivity, action models, number of objectives, and vulnerabilities
Automatic attack graph generation	Tools (AVAIN, Naggen, MulVal, PARMS, AVRA, IOT-PEN, GADG, and GRAVITAS). Graphs generated up to 1000 s of nodes, directed graphs without and with cycles
Properties analyzed	Properties captured in linear temporal logic of attack paths. Real-time network evaluation via dynamic analysis
Violations detected	Network paths resulting in exploits of individual hosts
Type of attacks	MQTT-based attacks, DDoS, RPL rank property vulnerabilities**,** SQL Injection, susceptible code clones, weak cryptographic algorithms, CVEs
Visualization	Tools (Apache Spark Databricks, Graphviz, and NetSpa), methodology (separating network-topology information and host-vulnerability information in the graph, ranking the nodes in the graph, creating subgraph).
IoT devices/systems	IP cameras, HDMI sticks, smartwatches, activity trackers, drones, temperature sensors, heart rate sensors, activity sensors, mobile phones, pacemakers, Industrial IoT networks, Power grids, and SCADA networks.
Derived recommendations	Cost versus benefits to breaking the attack paths. Determining the most effective and vulnerable hosts in the IoT network to secure. Identification of exploitable vulnerabilities in real-time.

quantify attack detection time in minutes or seconds [33], [34], [45], [46]. Consequently, one potential direction for additional research is to develop a framework that identifies exploitable vulnerabilities and to what extent they can be exploited in real-time [41].

To achieve optimal performance, attack graph solutions for the IoT should be designed for a specific system or application domain instead of being overly generic or abstract. For example, those aiming to generate or search for general exploits over many protocols or hosts may only detect on-the-surface violations or simple vulnerabilities [42]. Table 16 presents a summary of our findings, including the methods used, parameters represented, automatic attack graph generation tools, properties analyzed, types of attacks, visualization techniques, and IoT devices/systems used in existing research. This information can be used by practitioners to determine the best security solution and attack graph model to use for their specific IoT network requirements. The derived recommendations from the studies can help determine the most effective and vulnerable hosts in the IoT network to secure, identify exploitable vulnerabilities in real-time, and evaluate the cost versus benefits of breaking the attack paths. Overall, the paper provides a comprehensive analysis of the state of the art in attack graphs for IoT vulnerability assessment, and readers can refer to Table 16 to gain insight into the relevant information for individual requirements.

Furthermore, future attack graphs can be tested to assess multiple stages of their used models during the numerous vulnerability assessment stages (such as initial compromise versus post-exploitation using compact attack graphs). This is likely to enable more practical simulation and algorithmic learning. Furthermore, most reviewed studies use simulations or testbeds for evaluation [7], [13], [14], [15], [16], [20], [21], [23], [26], [29], [33], [35], [37], [39], [44], [51], [54], [67] instead of real-world IoT networks.

There is a knowledge gap related to using standardized IoT networks or systems that are sufficiently complex, realistic, and have parameters typical of numerous real-world applications. This will facilitate the design of a model to assess and benchmark the use of attack graphs for vulnerability assessment. The model assessment criteria could determine the effectiveness of graph theory in the vulnerability assessment of IoT networks.

Further, there is a need for a deep analysis of attack graph applications in the vulnerability assessment of a real-world IoT network. Most studies have used testbeds or simulations when applying attack graphs to assess the vulnerabilities of IoT networks. Developing an attack graph model for vulnerability assessment will be the future direction of this study.

The study presents a unique and comprehensive examination of the existing literature on attack graphs in the vulnerability assessment of IoT networks. Unlike previous studies that have demonstrated the use of attack graphs in IoT vulnerability assessment, the authors aim to provide a critical analysis of the state of the art in attack graph modeling and application in IoT networks, highlighting the strengths and weaknesses of previous studies. The study addresses limitations and gaps in previous studies by highlighting the challenges in the field and recommending new approaches. The study also provides new insights and results, such as the observation that Bayesian networks appear to have better results in solving uncertainties and association problems over time, and the recommendation that future IoT attack graphs should assess multiple stages of vulnerability assessment. The authors provide a valuable contribution to the existing body of knowledge in the field of attack graphs in IoT vulnerability assessment by presenting a comprehensive examination of key works, highlighting the strengths and weaknesses of previous studies, and presenting new insights and recommendations for future research in the field.

VIII. CONCLUSION

This study has investigated and reviewed the existing literature on attack graphs in the vulnerability assessment of IoT networks. The modeling of IoT systems as attack graphs provides an opportunity to analyze the system for security properties and weaknesses. As a methodology, attack graphs scale to model IoT systems (where the device units are highly heterogeneous) with valuable results. The current challenges are the dynamic nature of network attacks and the changing topology. To minimize real-time learning, vulnerability assessment using attack graphs has to be combined with complementary methodologies such as artificial intelligence, machine learning, and game theory. This would assist in obtaining a powerful system with significant interest. While the use of attack graphs in the vulnerability assessment of IoT networks has been demonstrated in prior studies, there is a need to keep pace with technological advancements and the evolution of cyberattack techniques to avoid violation detection. Based on the review, several observations have been made. Our study has determined that the scalability of attack graphs is one of the critical challenges, for instance, in approaches using POMDP attack graphs. The complexity, time, and resources involved in developing and evaluating the attack graphs of an enterprise-level IoT system pose problems. This research found other alternative methodologies, such as using a combination of network metrics. The metrics identified include network size, number of exposed hosts, genetic algorithms, network state, connectivity, action models, number of objectives, and vulnerabilities. As a result, it can be concluded that there is a need for more creative metrics that improve cybersecurity experts' ability to compare several attack graphs in mitigating attacks in complex industrial networks.

The study also suggests that the Bayesian networks appear to have better results in solving uncertainties and association problems over time. Furthermore, future IoT attack graphs should assess multiple stages of their used models during the numerous vulnerability assessment stages (such as initial compromise versus post-exploitation). Therefore, the study recommends that the simulations or testbed evaluations used to validate attack graph models should provide the precise and practical testing necessary for complex industrial IoT systems.

REFERENCES

[1] M. K. Hasan, M. Akhtaruzzaman, S. R. Kabir, T. R. Gadekallu, S. Islam, P. Magalingam, R. Hassan, M. Alazab, and M. A. Alazab, "Evolution of industry and blockchain era: Monitoring price hike and corruption using BIoT for smart government and industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 9153–9161, Dec. 2022, doi: 10.1109/TII.2022.3164066.

[2] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," *Knowl.-Based Syst.*, vol. 123, pp. 163–173, May 2017, doi: 10.1016/j.knosys.2017.02.023.

[3] R. Hassan, F. Qamar, M. K. Hasan, A. Hafizah, M. Aman, and A. S. Ahmed, "Internet of Things and its applications: A comprehensive survey," *Symmetry*, vol. 12, no. 10, p. 1674, 2020, doi: 10.3390/sym12101674.

[4] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decis. Support Syst.*, vol. 133, Jun. 2020, Art. no. 113303, doi: 10.1016/j.dss.2020.113303.

[5] K. Zhang and J. Liu, "Review on the application of knowledge graph in cyber security assessment," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 768, no. 5, Mar. 2020, Art. no. 052103, doi: 10.1088/1757-899x/768/5/052103.

[6] G. Yadav, K. Paul, A. Allakany, and K. Okamura, "IoT-PEN: An E2E penetration testing framework for IoT," *J. Inf. Process.*, vol. 28, pp. 633–642, Sep. 2020, doi: 10.2197/ips jip.28.633.

[7] R. Sahay, G. Geethakumari, and K. Modugu, "Attack graph—Based vulnerability assessment of rank property in RPL-6LOWPAN in IoT," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 308–313, doi: 10.1109/WF-IoT.2018.8355171.

[8] I. Hydara, A. B. M. Sultan, H. Zulzalil, and N. Admodisastro, "Current state of research on cross-site scripting (XSS)—A systematic literature review," *Inf. Softw. Technol.*, vol. 58, pp. 170–186, Feb. 2015, doi: 10.1016/j.infsof.2014.07.010.

[9] D. R. McKinnel, T. Dargahi, A. Dehghantanha, and K.-K.-R. Choo, "A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment," *Comput. Electr. Eng.*, vol. 75, pp. 175–188, May 2019, doi: 10.1016/j.compeleceng.2019.02.022.

[10] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009, doi: 10.1016/j.infsof.2008.09.009.

[11] S. Hosseini, B. Turhan, and D. Gunarathna, "A systematic literature review and meta-analysis on cross project defect prediction," *IEEE Trans. Softw. Eng.*, vol. 45, no. 2, pp. 111–147, Feb. 2019, doi: 10.1109/TSE.2017.2770124.

[12] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100219, doi: 10.1016/j.cosrev.2019.100219.

[13] N. Agmon, A. Shabtai, and R. Puzis, "Deployment optimization of IoT devices through attack graph analysis," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.*, May 2019, pp. 192–202, doi: 10.1145/3317549.3323411.

[14] X. Liu, "A network attack path prediction method using attack graph," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 1–8, Jun. 2020, doi: 10.1007/s12652-020-02206-5.

[15] G. George and S. M. Thampi, "A graph-based security framework for securing industrial IoT networks from vulnerability exploitations," *IEEE Access*, vol. 6, pp. 43586–43601, 2018, doi: 10.1109/ACCESS.2018.2863244.

[16] M. Barrere and E. C. Lupu, "Naggen: A network attack graph generation tool—IEEE CNS 17 poster," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 378–379, doi: 10.1109/CNS.2017.8228667.

[17] M. Ge, J. B. Hong, H. Alzaid, and D. S. Kim, "Security modeling and analysis of cross-protocol IoT devices," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 1043–1048, doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.350.

[18] V. L. Shivraj, M. A. Rajan, and P. Balamuralidhar, "A graph theory based generic risk assessment framework for Internet of Things (IoT)," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2017, pp. 1–6, doi: 10.1109/ANTS.2017.8384121.

[19] A. R. Chandan and V. D. Khairnar, "Security testing methodology of IoT," in *Proc. Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2018, pp. 1431–1435, doi: 10.1109/ICIRCA.2018.8597192.

[20] G. Chu and A. Lisitsa, "Penetration testing for Internet of Things and its automation," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1479–1484, doi: 10.1109/HPCC/SmartCity/DSS.2018.00244.

[21] W. Nichols, Z. Hill, P. Hawrylak, J. Hale, and M. Papa, "Automatic generation of attack scripts from attack graphs," in *Proc. 1st Int. Conf. Data Intell. Secur. (ICDIS)*, Apr. 2018, pp. 267–274, doi: 10.1109/ICDIS.2018.00050.

[22] 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, doi: 10.1109/ACCESS.2018.2805690.

[23] T. Zitta, M. Neruda, L. Vojtech, M. Matejkova, M. Jehlicka, L. Hach, and J. Moravec, "Penetration testing of intrusion detection and prevention system in low-performance embedded IoT device," in *Proc. 18th Int. Conf. Mechatronics Mechatronika (ME)*, 2018, pp. 1–5. Accessed: Apr. 16, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8624734

[24] A. T. Al Ghazo and R. Kumar, "Identification of critical-attacks set in an attack-graph," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2019, pp. 716–722, doi: 10.1109/UEMCON47517.2019.8993076.

[25] H. Asri, H. Mousannif, and H. Al Moatassime, "Reality mining and predictive analytics for building smart applications," *J. Big Data*, vol. 6, no. 1, pp. 1–25, Jul. 2019, doi: 10.1186/s40537-019-0227-y.

[26] C. Cai, Y. Zhang, Z. Wang, and C. Xue, "A new model for securing networks based on attack graph," in *Proc. IEEE 4th Int. Conf. Signal Image Process. (ICSIP)*, Jul. 2019, pp. 318–324, doi: 10.1109/SIPROCESS.2019.8868357.

[27] R. Egert, T. Grube, D. Born, and M. Muhlhauser, "AVAIN—A framework for automated vulnerability indication for the IoT in IP-based networks," in *Proc. Int. Conf. Netw. Syst. (NetSys)*, Mar. 2019, pp. 1–3, doi: 10.1109/NetSys.2019.8854493.

[28] A. Ibrahim, S. Bozhinoski, and A. Pretschner, "Attack graph generation for microservice architecture," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 1235–1242, doi: 10.1145/3297280.3297401.

[29] Y. Mathov, N. Agmon, A. Shabtai, R. Puzis, N. O. Tippenhauer, and Y. Elovici, "Challenges for security assessment of enterprises in the IoT era," Jun. 2019, *arXiv:1906.10922*. Accessed: Apr. 15, 2021.

[30] T. Musa, K. C. Yeo, S. Azam, B. Shanmugam, A. Karim, F. D. Boer, F. N. Nur, and F. Faisal, "Analysis of complex networks for security issues using attack graph," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2019, pp. 1–6, doi: 10.1109/ICCCI.2019.8822179.

[31] I. Nadir, Z. Ahmad, H. Mahmood, G. A. Shah, F. Shahzad, M. Umair, H. Khan, and U. Gulzar, "An auditing framework for vulnerability analysis of IoT system," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS PW)*, Jun. 2019, pp. 39–47. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8802414?casa_token=_btLEen4TOAAAAAA:LdVkSEpi3cc1Tlre_z_YH_of5Xg35aKdDFOkoVz0ljTtzNIgUAmmU9BVs0JhqaUujXSJmrTVVQ

[32] J. Payne, K. Budhraja, and A. Kundu, "How secure is your IoT network?" in *Proc. IEEE Int. Congr. Internet Things (ICIOT)*, Jul. 2019, pp. 181–188, doi: 10.1109/ICIOT.2019.00038.

[33] B. Yiğit, G. Gür, F. Alagöz, and B. Tellenbach, "Cost-aware securing of IoT systems using attack graphs," *Ad Hoc Netw.*, vol. 86, pp. 23–35, Apr. 2019, doi: 10.1016/j.adhoc.2018.10.024.

[34] J. Zeng, S. Wu, Y. Chen, R. Zeng, and C. Wu, "Survey of attack graph analysis methods from the perspective of data and knowledge processing," *Secur. Commun. Netw.*, vol. 2019, pp. 1–16, Dec. 2019, doi: 10.1155/2019/2031063.

[35] P. A. Abdalla and C. Varol, "Testing IoT security: The case study of an IP camera," in *Proc. 8th Int. Symp. Digit. Forensics Secur. (ISDFS)*, Jun. 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9116392

[36] O. Abu Waraga, M. Bettayeb, Q. Nasir, and M. A. Talib, "Design and implementation of automated IoT security testbed," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101648, doi: 10.1016/j.cose.2019.101648.

[37] A. T. Al Ghazo, M. Ibrahim, H. Ren, and R. Kumar, "A2G2V: Automatic attack graph generation and visualization and its applications to computer and SCADA networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3488–3498, Oct. 2020, doi: 10.1109/TSMC.2019.2915940.

[38] A. Chowdhary, D. Huang, J. S. Mahendran, D. Romo, Y. Deng, and A. Sabur, "Autonomous security analysis and penetration testing," in *Proc. 16th Int. Conf. Mobility, Sens. Netw. (MSN)*, Dec. 2020, pp. 508–515, doi: 10.1109/MSN50589.2020.00086.

[39] M. Ibrahim, A. Alsheikh, and A. Matar, "Attack graph modeling for implantable pacemaker," *Biosensors*, vol. 10, no. 2, p. 14, Feb. 2020, doi: 10.3390/bios10020014.

[40] D. Malzahn, Z. Birnbaum, and C. Wright-Hamor, "Automated vulnerability testing via executable attack graphs," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Security)*, Jun. 2020, pp. 1–10, doi: 10.1109/CyberSecurity49315.2020.9138852.

[41] V. Sachidananda, S. Bhairav, and Y. Elovici, "OVER: Overhauling vulnerability detection for IoT through an adaptable and automated static analysis framework," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 729–738, doi: 10.1145/3341105.3373930.

[42] G. Spanos, K. M. Giannoutakis, K. Votis, B. Viano, J. Augusto-Gonzalez, G. Aivatoglou, and D. Tzovaras, "A lightweight cyber-security defense framework for smart homes," in *Proc. Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2020, pp. 1–7, doi: 10.1109/INISTA49547.2020.9194689.

[43] O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, and A. Shabtai, "Extending attack graphs to represent cyber-attacks in communication protocols and modern IT networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 19, no. 3, pp. 1936–1954, May 2022, doi: 10.1109/TDSC.2020.3041999.

[44] R. Sumanth and K. N. Bhanu, "Raspberry Pi based intrusion detection system using K-means clustering algorithm," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Jul. 2020, pp. 221–229, doi: 10.1109/ICIRCA48905.2020.9183177.

[45] D. Ivanov, M. Kalinin, V. Krundyshev, and E. Orel, "Automatic security management of smart infrastructures using attack graph and risk analysis," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS4)*, Jul. 2020, pp. 295–300, doi: 10.1109/WORLDS450073.2020.9210410.

[46] U. Saxena, J. Sodhi, and Y. Singh, "A comprehensive approach for DDoS attack detection in smart home network using shortest path algorithm," in *Proc. 8th Int. Conf. Rel., Infocom Technol. Optim., Trends Future Directions (ICRITO)*, Jun. 2020, pp. 392–395, doi: 10.1109/ICRITO48877.2020.9197763.

[47] B. Burr, S. Wang, G. Salmon, and H. Soliman, "On the detection of persistent attacks using alert graphs and event feature embeddings," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–4, doi: 10.1109/NOMS47738.2020.9110439.

[48] W. Hu, L. Zhang, X. Liu, Y. Huang, M. Zhang, and L. Xing, "Research on automatic generation and analysis technology of network attack graph," in *Proc. IEEE IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2020, pp. 133–139, doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00033.

[49] B. Yuan, Z. Pan, F. Shi, and Z. Li, "An attack path generation methods based on graph database," in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Jun. 2020, pp. 1905–1910, doi: 10.1109/ITNEC48623.2020.9085039.

[50] V. Shakhov and I. Koo, "Graph-based technique for survivability assessment and optimization of IoT applications," *Int. J. Softw. Tools Technol. Transf.*, vol. 23, no. 1, pp. 105–114, Nov. 2020, doi: 10.1007/s10009-020-00594-9.

[51] M. Jang, D. Kim, D. Seo, Y. Ju, S. Ryu, and H. Yoon, "An intelligent recommendation algorithm for red team strategy in edge computing powered massive cyber defense exercise," *Comput. Commun.*, vol. 165, pp. 141–148, Jan. 2021, doi: 10.1016/j.comcom.2020.10.008.

[52] J. Jiao, H. Zhao, and H. Cao, "Using deep learning to construct auto web penetration test," in *Proc. 13th Int. Conf. Mach. Learn. Comput.*, Feb. 2021, pp. 59–66, doi: 10.1145/3457682.3457691.

[53] M. Chowdhury, B. Ray, S. Chowdhury, and S. Rajasegarar, "A novel insider attack and machine learning based detection for the Internet of Things," *ACM Trans. Internet Things*, vol. 2, no. 4, pp. 1–23, Jul. 2021, doi: 10.1145/3466721.

[54] A. Mudgerikar, P. Sharma, and E. Bertino, "Edge-based intrusion detection for IoT devices," *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 4, pp. 1–21, Dec. 2020, doi: 10.1145/3382159.

[55] Z. Zhang, J. Jia, B. Wang, and N. Z. Gong, "Backdoor attacks to graph neural networks," in *Proc. 26th ACM Symp. Access Control Models Technol.*, Jun. 2021, pp. 15–26, doi: 10.1145/3450569.3463560.

[56] X.-F. Wang, T.-Y. Zhou, and J.-H. Zhu, "Network security assessment based on full host-based attack graph," in *Proc. Int. Conf. Cyberspace Innov. Adv. Technol.*, Dec. 2020, pp. 230–235, doi: 10.1145/3444370.3444577.

[57] L. Gressl, C. Steger, and U. Neffe, "Design space exploration for secure IoT devices and cyber-physical systems," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 4, pp. 1–24, Jul. 2021, doi: 10.1145/3430372.

[58] E. Muhati and D. B. Rawat, "Hidden-Markov-model-enabled prediction and visualization of cyber agility in IoT era," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9117–9127, Jun. 2022, doi: 10.1109/JIOT.2021.3056118.

[59] W. Liu and T. Zhao, "Vulnerability assessment and attack simulation of power IoT based on the attractiveness of equipment assets," in *Proc. IEEE 4th Adv. Inf. Manage., Commun., Electron. Autom. Control Conf. (IMCEC)*, Jun. 2021, pp. 1246–1250, doi: 10.1109/IMCEC51613.2021.9482124.

[60] I. Stellios, P. Kotzanikolaou, and C. Grigoriadis, "Assessing IoT enabled cyber-physical attack paths against critical systems," *Comput. Secur.*, vol. 107, Aug. 2021, Art. no. 102316, doi: 10.1016/j.cose.2021.102316.

[61] C. Skandylas, L. Zhou, N. Khakpour, and S. Roe, "Security risk analysis of multi-stage attacks based on data criticality," in *Proc. IEEE/ACM 2nd Int. Workshop Eng. Cybersec. Crit. Syst. (EnCyCriS)*, Jun. 2021, pp. 13–20, doi: 10.1109/EnCyCriS52570.2021.00010.

[62] S.-Z. Liu, C.-W. Shao, Y.-F. Li, and Z. Yang, "Game attack–defense graph approach for modeling and analysis of cyberattacks and defenses in local metering system," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2607–2619, Jul. 2021, doi: 10.1109/TASE.2021.3093082.

[63] R. Maciel, J. Araujo, C. Melo, P. Pereira, J. Dantas, J. Mendonça, and P. Maciel, "Impact evaluation of DDoS attacks using IoT devices," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr./May 2021, pp. 1–8. Accessed: Jul. 28, 2021, doi: 10.1109/SysCon48628.2021.9447145.

[64] T. Setzler and X. Mountrouidou, "IoT metrics and automation for security evaluation," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2021, pp. 1–4. Accessed: Jul. 28, 2021, doi: 10.1109/CCNC49032.2021.9369533.

[65] R. He, X. Ji, and W. Xu, "Threat assessment for power industrial control system based on descriptive vulnerability text," in *Proc. IEEE 4th Conf. Energy Internet Energy Syst. Integr. (EI2)*, Oct. 2020, pp. 3844–3849, doi: 10.1109/EI250167.2020.9346835.

[66] J. Brown, T. Saha, and N. K. Jha, "GRAVITAS: Graphical reticulated attack vectors for Internet-of-Things aggregate security," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 3, pp. 1331–1348, Jul. 2022, doi: 10.1109/TETC.2021.3082525.

[67] R. Wu, J. Gong, W. Tong, and B. Fan, "Network attack path selection and evaluation based on *Q*-learning," *Appl. Sci.*, vol. 11, no. 1, p. 285, Dec. 2020, doi: 10.3390/app11010285.

[68] Y. Li and X. Li, "Research on multi-target network security assessment with attack graph expert system model," *Sci. Program.*, vol. 2021, pp. 1–11, May 2021, doi: 10.1155/2021/9921731.

[69] A. Alharbi and K. Alsubhi, "Botnet detection approach using graph-based machine learning," *IEEE Access*, vol. 9, pp. 99166–99180, 2021, doi: 10.1109/ACCESS.2021.3094183.

[70] E. R. Ling and M. Ekstedt, "Generating threat models and attack graphs based on the IEC 61850 system configuration description language," in *Proc. ACM Workshop Secure Trustworthy Cyber-Phys. Syst.*, Apr. 2021, pp. 98–103, doi: 10.1145/3445969.3450421.

[71] V. Shandilya, C. B. Simmons, and S. Shiva, "Use of attack graphs in security systems," *J. Comput. Netw. Commun.*, vol. 2014, pp. 1–13, Oct. 2014, doi: 10.1155/2014/818957.

[72] S. Noel, E. Harley, K. H. Tam, M. Limiero, and M. Share, "CyGraph: Graph-based analytics and visualization for cybersecurity," in *Handbook of Statistics*, vol. 35. Amsterdam, The Netherlands: Elsevier, 2016, pp. 117–167, doi: 10.1016/bs.host.2016.07.001.

[73] S. Siboni, V. Sachidananda, Y. Meidan, M. Bohadana, Y. Mathov, S. Bhairav, A. Shabtai, and Y. Elovici, "Security testbed for Internet-of-Things devices," *IEEE Trans. Rel.*, vol. 68, no. 1, pp. 23–44, Mar. 2019, doi: 10.1109/TR.2018.2864536.

[74] A. Tekeoglu and A. Tosun, "An experimental framework for investigating security and privacy of IoT devices," in *Proc. Int. Conf. Intell., Secure, Dependable Syst. Distrib. Cloud Environ.*, in Lecture Notes in Computer Science, 2017, pp. 63–83, doi: 10.1007/978-3-319-69155-8_5.

OMAR SAIF MUSABBEH BIN HAMED ALMAZROUEI received the B.Sc. degree from the University of Utah, USA, and the master's degree in cybersecurity from Zayed University, United Arab Emirates, in 2012. He is currently pursuing the Ph.D. degree with Universiti Teknologi Malaysia. He is a Cybersecurity Expert with Universiti Teknologi Malaysia. His research interests include penetration testing and reverse engineering.

PRITHEEGA MAGALINGAM received the Doctor of Philosophy (Ph.D.) degree from the Royal Melbourne Institute of Technology, Melbourne, Australia, in 2015. She is currently a Senior Lecturer with the Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur, and a certified HRD Trainer. She has nine years of experience in building new tools using R for network science applications. Recently, she is also exploring and expanding her research in business intelligence and analytics using complex network analysis. She has published in several refereed journals, such as *Digital Investigation*, *Journal of Telecommunication, Electronic and Computer Engineering*, *International Journal of Advanced Computer Science and Applications*, *Journal of Physics: Conference Series*, *Computer Science Review*, *British Food Journal*, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and *Journal of Information Technology Management*. Her research interests include the applications of complex network analysis to build a new approach to detecting a criminal community, digital forensics analysis and validation, data analytics, and information security risk management. She was a member of the IEEE Information Theory Society and Computer Society, in 2016, and the Information Systems Audit and Control Association (ISACA), in 2017.

MOHAMMAD KAMRUL HASAN (Senior Member, IEEE) received the Doctor of Philosophy (Ph.D.) degree in electrical and communication engineering from the Faculty of Engineering, International Islamic University, Malaysia, in 2016. He is currently a Senior Lecturer of network and communication technology research cluster with the Center for Cyber Security, Universiti Kebangsaan Malaysia (UKM). He is specialized with elements pertaining to cutting-edge information centric networks, computer networks, data communication and security, mobile network and privacy protection, cyber-physical systems, the Industrial IoT, transparent AI, and electric vehicles networks. He has published more than 150 indexed papers in ranked journals and conference proceedings. He is a member of Institution of Engineering and Technology and the Internet Society. He is a certified Professional Technologist (P.Tech./Ts.) in Malaysia. He also served as the Chair for the IEEE Student Branch, from 2014 to 2016. He has actively participated in many events/workshops/trainings for the IEEE and IEEE humanity programs in Malaysia. He is an Editorial Member in many prestigious high-impact journals, such as IEEE, IET, Elsevier, Frontier, and MDPI. He is the general chair, the co-chair, and a speaker for conferences and workshops for the shake of society and academy knowledge building and sharing and learning. He has been contributing and working as a volunteer for underprivileged children for the welfare of society.

MOHANA SHANMUGAM received the Ph.D. degree in information systems from Universiti Putra Malaysia, in 2017. She is currently a certified PSMB Trainer and a Senior Lecturer with the Informatics Department, College of Computing and Informatics, Universiti Tenaga Nasional UNITEN. She has more than 14 years of experience handling a diverse academic process caseload and more ten years of experience as an active researcher. She has published in several refereed journals, such as *Journal of Technological Forecasting and Social Change*, *Journal of Enterprise Information Management*, *International Journal of Information Management*, *Informatics for Health and Social Care*, and *Journal of Business Research*. Her research interests include social commerce, IS theories, social media marketing, and IS adoption.