Cyber Deception Across Domains: A Comprehensive Survey of Techniques, Challenges, and Perspectives

Amal Sayari, Slim Rekhis
University of Carthage, Higher School of Communication of Tunis (SUP'COM),
LR11TIC04, Communication Networks and Security Research Lab. & LR11TIC02,
Green and Smart Communication Systems Research Lab, Tunisia

***Abstract*—Cloud environments (CE), wireless networks (WN), cyber-physical systems (CPS), industrial control systems (ICS), smart grids (SG), internet of things (IoT), internet of vehicles (IOV), and unmanned aerial vehicles (UAV), are currently popular targets for cyberattacks due to their inherent limitations and vulnerabilities. Each domain has its own attack surfaces, weaknesses, and areas for implementing defense strategies appropriate to its specific conditions. Among the various defense mechanisms discussed in previous years, cyber deception has appeared as a very promising method. This approach allows the defenders to steer the attackers in the wrong direction, get threat intelligence, and at the same time, increase security by engaging with adversaries in deception environments in a proactive manner. Cyber deception has been a topic of investigation in several studies, where specific frameworks and techniques were proposed to identify, delay, or disrupt adversarial behavior. Nevertheless, the contributions of earlier works are frequently limited or missing a unified framework that makes a thorough and comparative study necessary. This survey investigates the cyber deception techniques used in various domains. The first part is about the cores of deception and its background. Next, it presents a summary of the available deception techniques with their modeling by different frameworks like MITRE ATT&CK, D3FEND, and Engage, and intelligent orchestration using reinforcement learning (RL) and game theory (GT). Then, it serves as a thorough systematic review of each selected paper, going over the system design, used deception techniques, evaluation metrics, and limitations on each scheme. The achieved results are compiled into a unified summary table to enable a quick and effective comparison across the domains. It concludes, therefore, by discussing the main challenges, open issues, and areas of research that have not yet been explored, thus making it a valuable source for future research on cyber deception.**

***Keywords*—Cyber defense; cyber deception; cloud environments; wireless networks; cyber-physical systems; industrial control systems; smart grids; internet of things; internet of vehicles; unmanned aerial vehicles**

I. Introduction

The growth of cyber-attacks in terms of frequency and sophistication has outstripped the ability of conventional security measures to be the sole protector of the complex and dynamic digital infrastructures. The adversaries are continually adapting their tactics, techniques, and procedures (TTPs) to evade traditional defense mechanisms like firewalls, intrusion detection/ prevention systems, and antivirus software. The reactive schemes act typically after the initiation of an attack, makes the need for different, proactive, and intelligent defense strategies even more urgent.

One promising strategy is cyber deception, which is currently widely known in both research and industry. Cyber deception is putting uncertainty and misdirection in the environment by intentionally misguiding adversaries; thus, their progression becomes slower or they get fictitious information. The method is based on failure or turning away the attacker, providing first warning through the engagement of the attacker, and collecting information about their capabilities. Deception methods, from honeypots and honeynets to honeytokens [1], are successful complementary tools that shift the advantage back to defenders.

At the same time, the wide digital landscape has extended to encompass a wide variety of interconnected and heterogeneous systems, including CE, WN, CPS, ICS, SG, IOT, IOV, and UAV. Apart from the fact that the aforementioned domains have achieved a previously unattainable level of scalability and interconnection, they are also the main sources of new vulnerabilities. Each domain, besides the attacks, also comes with its unique features such as different architectural structures, operational constraints, communication protocols, and security requirements that determine the nature and intensity of the cyber threats that it has to deal with.

Although the individual fields such as WN, CPS, or IOT are seeing a significant increase in the use of cyber deception techniques, the current literature is still fragmented. At present, there is no unified study that examines how deception has been adapted and applied across such diverse technological environments. In particular, research on honeypots and honeynets tends to remain siloed, without much cross-domain analysis and integration.

Previously conducted surveys on cyber deception [2] [3] [4] [5] [6] [7] have introduced foundational taxonomies, described deception techniques, and outlined conceptual models. However, those are mainly dedicated to technique-level analysis (e.g., GT, RL, or Honey-X strategies) and they are not systematic comparing across different operational domains. Although some works [4] [6] did provide specific domains (e.g., in IoT, ICS, CPS, and CE), the exploration is still fragmented, and it has not been focused enough on how architectural constraints, attacker models, and environmental characteristics influence the design and deployment of deception strategies.

Based on those limitations, we have prepared a thorough and comparative review of the cyber deception methods in several critical domains. In contrast to previous studies that either dwell upon particular techniques or theoretical models, our survey stresses the domain-aware analysis of deception

strategies, elaborating on customization to different system architectures and operational contexts. Here are our main contributions:

- We present the fundamental concepts, models, and strategies of cyber deception to provide a clear background for the survey.
- We meticulously extract and synthesize the deception technique, architecture, evaluation metrics, and limitations from each selected paper.
- We categorize the findings by domain and summarize them in standardized tables for cross-domain comparison.
- We identify common challenges, open issues, and trends, which could serve as a guide for further research in deception-based cyber defense in various contexts.

The remainder of this survey is structured as follows. Section II presents the adopted methodology. Section III addresses a comparison with existing surveys. Section IV provides the background and key concepts of cyber deception. Section V describes the intelligent orchestration of cyber deception. Sections VI to XIII present the exploration of cyber deception in CE, WN, CPS, ICS, SG, IoT, IoV, and UAV, respectively. Section XIV discusses open research issues and key insights. Finally, Section XV concludes the study.

II. Adopted Methodology

A systematic literature review was performed to investigate the use of cyber deception techniques in various domains. The review process was structured in a multi-step manner to achieve comprehensiveness, relevance, and quality.

1) Database search and initial collection: We queried high-impact databases like IEEE Xplore, DBLP, and Google Scholar using various combinations of both general and domain-specific keywords such as “cyber deception”, “honeypot”, “honeytoken”, “fake data”, and other domain-specific terms like “IoT”, “ICS”, “UAV”, “Smart Grid”, and “Cloud”.
2) Inclusion and exclusion criteria: Only the articles that described defensive deception techniques were kept. Articles that were about offensive deception techniques were removed from the list. Besides that, we also filtered out duplicates and inaccessible entries.
3) Reference expansion and survey integration: To make sure all important works were included, we examined and integrated the previous surveys along with their key referenced papers. Each past survey was examined to find the gaps, list the most important works, and to make sure there were no overlaps. The articles that were already included in the previous surveys were also taken into account in order to ensure continuity and progression.
4) Screening and filtering: We did the screening of titles and abstracts first and made the full-text reviews to confirm the relevance of the articles to cyber deception. During this process, the studies were indicated according to the techniques used for deception, the target domain, the architectural framework adopted, evaluation metrics, and the limitations identified. Thus, ∼140 papers were selected.
5) Domain relevance-based filtering: We pursued a filtering strategy based on the articles’ dates of publication that would be on a period extending approximately six years (2019 to 2024), alongside a selective insertion of emerging contributions from 2025. Then, from the set of ∼140 papers, 46 papers were selected as relevant to cyber deception exploration by domain, as illustrated in Fig. 1. The others were used to properly define the cyber deception concepts, techniques, and models.

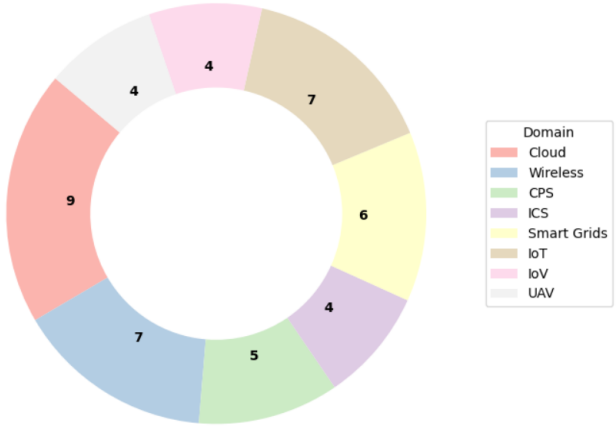

Fig. 1. Distribution of selected papers by domain.

III. Comparison with Existing Surveys

In [2], the authors explored the classification of cyber deception, which cuts across four dimensions: the deceit unit, application layer, defense goal, and deployment mode. They presented techniques like honeypots, honeyfiles, and fake topologies across the network, system, application, and data layers. They further outlined the deployment models as process-based, probabilistic, graph-based, and game-theoretical approaches. The major challenges that were mentioned are the evaluation of deception effectiveness and the automation of dynamic deployment in complicated environments.

In [3], the authors explored 24 studies regarding the application of game theory in the representation of cyber deception and created a taxonomy that contains six types of deception: perturbation, moving target defense (MTD), obfuscation, mixing, honey-x, and attacker engagement. The taxonomy is based on game-theoretic components, including agents, actions, and information structure. Specific deception types were related to the appropriate models, such as stackelberg, Nash, and signaling games. Furthermore, the research gaps were identified by the authors as mimetic deception and practical implementation.

In [4], the authors put forth a completely different classification framework to use for cyber deception based on the following features: conceptual categories, physical and virtual artifacts, intended effects, goals (protection or detection), and

deception activeness. They used game theory (GT) and machine learning (ML) for the discussion of deception techniques, as they pointed out GT's strategic modeling and ML's feature of generating realistic decoys. Their review was complemented by applications in different settings such as CPS, cloud, IoT, and software-defined networking (SDN), and they performed an analysis of evaluation metrics and testbeds. They ended their work by pointing out the existing limitations and urging further exploration of the combination of GT and ML for more adaptive deception strategies.

In [5], the authors made a review of different frameworks of cyber deception in terms of network security, assessing their strengths and weaknesses. They declared that the most effective deception requires network topologies that can adapt, which include scalability, dynamic reconfiguration, and fast responses. They pointed out SDN is the principal facilitator to the deployment of various techniques of deception such as honeypots, honeynets, and Moving Target Defense. Moreover, they assigned the challenges of real-time agility, DDoS resilience, and user impact minimization, and offered directions for the future, such as ML-driven adaptive deception and enhanced network virtualization for APT defense.

In [6], the authors were engaged in scrutinizing Honey-X-techniques such as honeypots, honeynets, honeytokens, and honeywebs. Besides, they also got acquainted with the mechanisms of defense like MTD, ML, and GT, and their works in synergy with them. They studied Honey-X- based strategies in different sectors such as ICS, IoT, critical infrastructure, and web applications, stressing the fact that they can be used in different environments. The effectiveness of these strategies was calculated through metrics such as Mean Time to Compromise (MTTC) and Mean Time to System Failure (MTTSF). The final point they made was the presentation of key research challenges, such as improving honeypot realism and scalability, deception placement optimization through GT, and combining MTD with ML for enhanced threat detection and response.

In [7], the authors investigated deception techniques conceived to boost honeypots and honeynets. These techniques include advanced mimicking, manipulative cooperation, fake databases, honeytokens, and traffic redirection. For this purpose, authors have proposed a categorization system for honeypots based on the following criteria: purpose, interaction level, implementation type, and activity level. They also shared a mathematical model that is supposed to aid in the optimization of both the configuration and the deployment. The evaluation of their approach was realized by comparison with other approaches in terms of deception metrics like discrepancy, wasted time, and adversaries that were returned. The shortcomings they uncovered were in areas such as dynamic, 5G, and SDN-based honeypots, and they suggested future research on ML-driven honeypot optimization and also on more advanced deception strategies.

IV. Background on Cyber Deception

A. Key Concepts of Cyber Deception

Deception originated in military contexts, where it is used to mislead adversaries regarding one's strengths, weaknesses, intentions, and tactics [8]. This tactic includes forms of battlefield deception like camouflage, feints (pretend attacks), ruses (tricks), demonstrations (fake force deployment), and displays (like showing fake military equipment, for example, inflatable tanks). This concept has extended into the cyber domains as cyber deception, refined by Almeshekah and Spafford [9] as "planned actions taken to mislead and/or confuse attackers and to thereby cause them to take (or not take) specific actions that aid computer security defense", and categorized by Whaley [10] into two taxonomies: dissimulation and simulation.

Dissimulation or hiding is a trick of information or system states to prevent hackers from understanding or discovering the true nature of the system. And the purpose of this idea is to mislead hackers or unauthorized users by showing incorrect information, thus making it difficult for them to guess what they should decide. This method is opposite to the traditional protection mechanisms like firewalls, access control, and encryption that mainly resist by prohibiting access or concealing data rather than by misleading the attacker. Simulation or showing is concerned with the making of environments, systems, services, and data that look authentic and operational and thus deceive the attacker into thinking he is dealing with real assets. Such defensive environments are meant to deceive threats, involve enemies, and watch their actions and designs, all without paving the way for real systems or data to be exposed. Simulated components are off the grid yet unbelievably convincing. For instance, after being locked out of the system several times, the system could keep showing the login prompts without ever giving the user permission to log in. The hacker would then think that they still have a chance to enter the system.

Dissimulation and simulation can be used either separately or cooperatively, in accordance with the objectives of the strategy. Dissimulation is the primary means of reducing one system's visibility and appeal, particularly in environments focused on confidentiality and stealth. On the other hand, simulation is the best solution for threat detection by inducing adversaries into a false environment. Simultaneously acting, these two methods create total deception of the mechanism that both distracts and engages adversaries to be able to protect from and gather credible information on the threat.

One of the instances of using deception techniques for cybersecurity can be found in Cliff Stoll's book "The Cuckoo's Egg" [11], where he created a fake system environment with a user account containing fake documents to deceive and stall attackers in order to track their activities and reveal their identities. In the 2000s, honeypots became an adopted method for observing, analyzing, understanding, and modeling attackers' actions [12] [13]. Subsequently, various honey-related approaches, called honeytokens [1] have emerged. These deception techniques fall under six principal tactics [14]. Three of them fall under dissimulation: masking (conceals real systems, data, or network attributes to prevent detection or mislead attackers about their true nature), repackaging (alters the appearance or structure of data or code, such as changing formats or using wrappers, to evade detection and analysis), and dazzling (generates excessive noise such as fake traffic, logs, or alerts to obscure valuable information and confuse attackers). The other three tactics fall under simulation: mimicking (imitates the behavior of real systems,

users, or data to make deceptive assets appear authentic and lure attackers), inventing (creates entirely fictitious systems, data, or user identities that have no real counterpart, designed to mislead and trap attackers), and decoying (deploys simulated assets such as honeypots, honeynets to attract, engage, and analyze attackers while protecting real systems).

Moreover, deception techniques can be leveraged to achieve four principal cybersecurity goals [9]: detection (help identify malicious activity where any interaction is inherently suspicious, by spamming fake assets), prevention (attackers can be deterred or slowed down using creating uncertainty and increasing the perceived risk of being detected), response (deceptive systems can be securely detached and examined post-attack, thus, making the process of incident response and forensic analysis efficient), and research (provide regulated settings to examine the behavior of attackers, create threat intelligence, and study newly developed malware).

B. Layer-Based Cyber Deception Techniques

For the sake of generality and consistency across domains, we adopt the classification proposed by the authors in [2] and further detailed by the authors in [15], which categorizes deception techniques into a four-layer deception stack:

- Network-based deception techniques: are made to deceive attackers in the reconnaissance and exploitation phases that involve manipulation of the network-level information. Examples include altering network topology [16] [17] [18], randomizing IP addresses [19] [20] [21], tarpits [22] [23], deploying network honeypots [24] [25] [26] [27], and using honeyports [28] [29]. Such strategies delay or misguide attackers by presenting a false view of the network infrastructure, thereby increasing their time, cost, and reducing the effectiveness of their probing activities.
- System-based deception techniques: focus on deceiving adversaries within the host or operating system environment. Techniques include honeypatches [30] [31], Ghost Patches [32] [33], deceptive OS fingerprinting [34] [35], deceptive system calls [36][37], and honey RAM / memory injection [38], which allow defenders to monitor attacker behavior without exposing real systems.
- Application-based deception techniques: operate at the software or application interface layer, where user or attacker interaction takes place. They include honeypermissions [39], honeyaccounts [28], honeyprofile (OSN) [40] [41], honeyemail [42], decoy web forms/ fields [43], delayed response (fake feedback) [44] [45], fake API endpoints [46], and decoy hyperlinks [47] [48]. Such methods are effective at detecting and delaying attackers during application-level attacks like SQL injection, credential theft, or web exploitation.
- Data-based deception techniques: target the attacker's goal, access to sensitive data, by injecting deceptive or false data into the system. Examples include decoy document [49], honey URL [50] [51], honeyentries [52] [53] [54], honeyfiles [55] [56] [57] [58], honeywords [59] [60] [61] [62], honey encryption [63], honey database/ metadata [64] [65], and decoy source code [66], all designed to mislead, trace, or waste the resources of attackers attempting to exfiltrate valuable information.

The placement of these deception techniques within a target domain can either be a part of the existing security solution or can be isolated and used as a standalone mechanism. In our classification, we adopt the logic explored in [2], which defines four main deployment modes based on security objectives, system architecture, and operational constraints:

- Built-in deception/ part of: embedded directly into the system during the design phase, such as deceptive logic in source code or modified system responses.
- Add-on deception/ part of: incorporated into an operational system at runtime, through the insertion of honeytokens, fake files, or configuration traps.
- In-front deception/ part of/ intermediary: positioned as a proxy or gateway in front of the real system to intercept and manipulate incoming interactions, allowing early engagement with potential threats.
- Isolated deception/ standalone: deployed in separate, decoupled environments such as honeynets, decoy servers, that mimic real systems to lure attackers and analyze their behavior without impacting production assets.

In Table I, we provide a consolidated overview of representative cyber deception techniques, highlighting their core characteristics in terms of taxonomy, tactics, goals, application layers, and deployment modes as extracted from the surveyed literature. The majority of the cyber deception techniques outlined are designed for dual purposes, simulation and dissimulation, because building false artifacts while concealing true ones at the same time is the precondition of effective deception in most cases. The dual effectiveness increases both the credibility and overall impact of that deception. Also, a lot of the techniques can be utilized on different layers, such as network, system, and application, as they are designed to be adaptable and provide flexibility in their implementation. Therefore, the same technique may cover different security objectives, such as prevention, detection, response, and research, based on how and where it is implemented.

A multi-layer deception system is depicted in Fig. 2. It is a combination of some of the techniques that have been previously discussed. The multi-layer architecture, comprising the network, system, application, and data layers, encompasses both real and honey elements. Each of the honey components (e.g., honeyserver, honeyfile, honeydatabase) imitates a corresponding real counterpart (e.g., server, file, database) to mislead and monitor the attacker's behavior. The analyst server functions as a pivotal element by collecting alerts and logs from all honey elements, which provide a basis for detection, analysis, and response to malicious activities triggered by interactions with the deceptive assets.

C. Key Evaluation Metrics for Cyber Deception

To evaluate the efficiency of cyber deception strategies, researchers have suggested a variety of quantifiable metrics

TABLE I. OVERVIEW OF CYBER DECEPTION TECHNIQUES

Deception technique	Ref	Taxonomy	Tactic	Goal	Layer	Deployment mode
Altering Network Topology	[16]	Simulation + Dissimulation	Inventing, Dazzling	Prevention, Detection	Network	In-front
Reconnaissance Deception System (RDS) using Virtual Network Views	[17]	Simulation + Dissimulation	Inventing, Dazzling, Decoying	Detection, Prevention	Network	In-front, Add-on, Isolated
Virtual Network Topology Deception Defense (VNTDD)	[18]	Simulation + Dissimulation	Inventing, Dazzling, Decoying	Detection, Prevention	Network	In-front, Add-on, Isolated
Randomizing IP Addresses	[19]	Simulation + Dissimulation	Dazzling, Mimicking, Decoying	Detection, Prevention	Network	In-front, Add-on, Isolated
	[20]	Simulation + Dissimulation	Dazzling, Mimicking	Detection, Prevention	Network	Add-on, In-front
Path Randomization	[20]	Simulation	Dazzling	Prevention	Network	Add-on, Isolated
Flexible Random Virtual IP Multiplexing (FRVM)	[21]	Simulation + Dissimulation	Dazzling, Mimicking	Prevention, Detection	Network	Add-on, In-front
Tarpits	[22]	Simulation + Dissimulation	Dazzling, Decoying	Detection, Prevention	Network	Add-on, Isolated
	[23]	Simulation + Dissimulation	Dazzling, Decoying, Mimicking	Detection, Prevention, Research	Network	Add-on, Isolated, In-front
Honeypot	[24]	Simulation + Dissimulation	Decoying, Inventing	Detection, Prevention, Response	Network	Add-on, In-front, Isolated
	[25]	Simulation + Dissimulation	Decoying, Mimicking	Detection, Prevention, Research	Network, Application	Add-on, In-front, Isolated
Network Honeypots	[26]	Simulation + Dissimulation	Inventing, Decoying, Dazzling	Detection, Prevention	Network, Application	Add-on, Isolated
Modular High-Interactivity Honeypot System	[27]	Simulation + Dissimulation	Inventing, Decoying, Dazzling, Mimicking	Detection, Prevention, Response, Research	Network, System, Application	Add-on, Isolated, In-front
Honeytoken-based deception framework	[28]	Simulation + Dissimulation	Inventing, Decoying, Masking, Repackaging	Detection, Response	System, Application, Data	Add-on, Isolated
HoneyPort (meta-honeypot system)	[29]	Simulation + Dissimulation	Inventing, Decoying, Mimicking, Dazzling	Detection, Prevention, Research	Network, System	Add-on, Isolated, In-front
Honeypatches	[30]	Simulation + Dissimulation	Mimicking, Decoying, Dazzling	Detection, Prevention, Response, Research	Application, System	Add-on, In-front, Isolated
	[31]	Simulation + Dissimulation	Mimicking, Decoying, Dazzling	Detection, Prevention, Response, Research	System, Application	Add-on, Isolated, In-front
Ghost Patches	[32]	Simulation + Dissimulation	Mimicking, Decoying, Dazzling	Detection, Prevention, Research	System, Application	Add-on, Isolated, In-front
Deceptive Patch Model (Faux, Obfuscated, Active Response Patches)	[33]	Simulation + Dissimulation	Mimicking, Inventing, Decoying, Dazzling, Repackaging	Detection, Prevention, Response, Research	System, Application	Add-on, In-front, Isolated
Deceptive OS Fingerprinting	[34]	Simulation + Dissimulation	Mimicking, Dazzling, Repackaging	Detection, Prevention, Research	Network, System, Application	Add-on, In-front, Isolated
	[35]	Simulation + Dissimulation	Dazzling, Mimicking, Repackaging	Detection, Prevention, Research	Network, System	Add-on, In-front, Isolated
Deceptive System Calls	[36]	Simulation + Dissimulation	Mimicking, Masking, Dazzling	Detection, Prevention	System	Add-on, Isolated
Honey RAM / Memory Injection	[37]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	System	Add-on, Isolated
	[38]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	System	Add-on, Isolated
Honeypermissions	[39]	Simulation + Dissimulation	Inventing, Masking	Detection, Prevention	Application	Add-on, Isolated
HoneyID	[28]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	Application	Add-on, Isolated
Honeyprofile (OSN)	[40]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	Application	Add-on, Isolated
	[41]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	Application	Add-on, Isolated
Honeyemail	[42]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	Application	Add-on, Isolated
Decoy Web Forms/Fields	[43]	Simulation + Dissimulation	Inventing, Mimicking, Decoying	Detection, Prevention	Application	Add-on, Isolated
Delayed Response (Fake Feedback)	[44]	Simulation + Dissimulation	Mimicking, Decoying, Inventing	Detection, Prevention	Application	Add-on
	[45]	Simulation + Dissimulation	Dazzling, Masking	Detection, Prevention	Application	Add-on, In-front
Fake API Endpoints	[46]	Simulation	Inventing, Decoying, Masking	Detection	Application	Add-on
Decoy Hyperlinks	[47]	Simulation + Dissimulation	Inventing, Decoying, Masking	Detection	Application	Add-on, In-front
	[48]	Simulation + Dissimulation	Inventing, Decoying, Masking	Detection	Application	Add-on, In-front
Decoy Documents	[49]	Simulation + Dissimulation	Inventing, Mimicking, Masking	Detection, Prevention	Data	Add-on, Isolated
Honey URL	[50]	Simulation + Dissimulation	Inventing, Masking	Detection, Response	Data	Add-on, Isolated
	[51]	Simulation + Dissimulation	Inventing, Masking	Detection, Response	Data	Add-on, Isolated
Honeyentries	[52]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Response	Data	Add-on, Isolated
	[53]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Response	Data	Add-on, Isolated
	[54]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Response	Data	Add-on, Isolated
Honeyfiles	[55]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Response	Data	Add-on, Isolated
	[56]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Response	Data	Add-on, Isolated
	[57]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Response	Data	Add-on, Isolated
	[58]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Research	Data	Add-on, Isolated
Honeywords	[59]	Simulation	Inventing	Detection, Prevention	Data	Add-on
	[60]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Prevention	Data	Add-on, Isolated
	[61]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Prevention	Data	Add-on, Isolated
	[62]	Simulation + Dissimulation	Inventing, Masking, Repackaging	Detection, Prevention	Data	Add-on, Isolated
Honey Encryption	[63]	Simulation + Dissimulation	Inventing, Repackaging, Masking	Detection, Prevention, Research	Data	Add-on, Isolated
Honey Database/Metadata	[64]	Simulation + Dissimulation	Inventing, Repackaging, Masking	Detection, Prevention	Data	Add-on, Isolated
Fake Document Infilling (FDI)	[65]	Simulation + Dissimulation	Masking, Repackaging, Mimicking	Detection, Prevention, Research	Data	Add-on, Isolated
Decoy Source Code	[66]	Simulation + Dissimulation	Inventing, Repackaging, Masking	Detection, Prevention, Research	Data	Add-on, Isolated

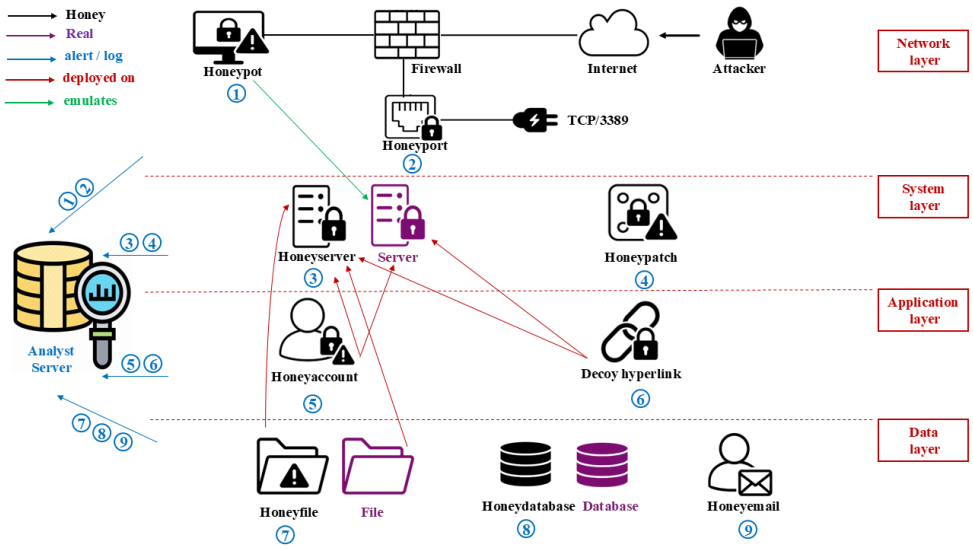

Fig. 2. An example of a multi-layered cyber deception architecture.

that can represent different components of system resilience, attacker behavior, and defense performance. The performance of the system and the progress of the attacker are evaluated with the help of temporal metrics like the Mean Time to Security Failure (MTTSF) [67] which symbolizes the period during which the system remains secure before it reaches a predefined security failure threshold due to a successful compromise, and the Mean Time to Compromise (MTTC) [67] where it represents the time required by the attacker to add a successful compromise to the network. Moreover, the metric of Vulnerability Analysis Time [68] assists in quantifying the amount of time the attacker has to spend identifying and then exploiting weaknesses present in the system, that is, from the complexity and skill level of each attacker. On the contrary, the Decision and Deployment Time (DDT) [69] is meant to show the period of time that passes between making a defensive decision and implementing it, such as deploying a honeypot.

Defense effectiveness is best evaluated through the use of probability-based indicators like the Defense Success Probability (DSP) [70] and the Success Rate of Attacks [71], which act as the basic tools to measure the quantitative success or failure of attaching defense devices under controlled experimental conditions. The measurement of detection metrics also focuses on Detection Accuracy [72], which is typically expressed in terms of the AUC of the ROC curve to judge the correct value of identifying malicious behavior. The other metric is the Mean Time to Detect Attacks [73], which is used for the evaluation of the efficiency of the deception-based detection system. In game-theoretic models of deception, the concept of Utility (or Payoff) is used most often to express the cost-benefit balance for both attacker and defender [74]. Moreover, the Attack Cost [75] evaluates the effort or risk that the attacker gets, including the chance of getting caught or wasted resources.

Resilience and operational continuity are represented by metrics such as Survival Rate [69], which refers to the ratio of the legitimate system components that have remained intact during attacks, and Round-Trip Time [69], which is the time taken by deception mechanisms (for example, code obfuscation or network slowdowns) to delay the attackers' progress.

D. Cyber Kill Chain for Phase-aware Deception Planning

The effectiveness of cyber deception strategies lies in the appropriate selection and positioning of the described deception techniques, which are customized to the particular-domain environment (e.g., UAV, IoT, ICS, Cloud) and the threat actors' behavior. Instead of randomly inserting deceptive assets, a threat-driven modeling approach is a must to prove that both deception strategies are useful and impactful. Two pertinent frameworks uphold this goal: the Cyber Kill Chain (CKC) and the MITRE ATT&CK framework. By using them together, they provide adversary tactics and lifecycle progression, and thus, an integrated view that helps in planning the optimization of deception for when, where, and how to deploy it.

The Cyber Kill Chain (CKC), originally introduced by Lockheed Martin, divides cyber-attacks into sequential phases: reconnaissance, weaponization, delivery, exploitation, installation, command and control (C2), and actions on objectives. This phase-based structure aids deception planning by helping defenders: i) determine the optimal timing for deploying deception (e.g., before delivery versus after exploiting), ii) align techniques to specific stages, for instance, deceptive network banners during delivery or honeyfiles during exfiltration, and iii) layer deception strategically across the kill chain, like using honeypots for exploitation, MTD for delivery, and fake data for exfiltration. Following these lines, the authors in [1] suggested an enhanced CKC model, meant to be used for deception and formed of three sub-kill chains: external, internal, and target manipulation. This refined model not only identifies the multi-stage progression of modern intrusions but also serves as a tool for mapping deception techniques (e.g., honeypots, honeytokens, MTDs) to both the attack phase and the affected system layer (network, system, software, data). By incorporating MITRE ATT&CK tactics into this structure, the model acts as an extensive framework for the orchestration of deception throughout the whole attack lifecycle.

E. MITRE ATT&CK for Tactic-technique Driven Deception

The MITRE ATT&CK [76] framework is based on the tactic-technique-procedure (TTP) representation of the behavior of adversaries, that is firmly based on the real-life observations on a wide range of platforms, namely, Enterprise, ICS, Mobile, and Cloud. ATT&CK plays a significant role in providing defenders with the following benefits: i) modeling attacker behavior by domain: for example, the ATT&CK for ICS stressed techniques such as "Inhibit Response Function" and "Manipulation of Control" which can also be found in the cyber-physical systems (CPS) and smart grid paradigms, ii) identifying deception targets: techniques like "Remote System Discovery", "Command and Scripting Interpreter", and "Credential Dumping" can be countered with honey APIs, honey credentials, or deceptive command environment, iii) building attack graphs: authors of [77] illustrated how techniques can be linked by logical preconditions and thus form attack paths. The graphs assist the defenders to optimize the positioning of decoys by focusing on the nodes that are most likely to be compromised with the least amount of resources.

In [78], the authors suggested a multi-layer graph-based way to model cyber-attacks and strategically allocate deception resources in alignment with threat modeling principles inspired by MITRE ATT&CK. The integration of network, service, and attack layers makes possible the detailed mapping of adversarial paths and their dependencies across system components. The model is capable of risk-based prioritization and preemptive allocation of deception assets such as honeypots and honeytokens at critical nodes. These strategies will contribute to the re-routing of attack chains and misleading of adversaries. This type of planning also drives the threat-informed deception by making the placement of the deception close to the attacker's behavior and technique paths that are illustrated in frameworks like MITRE ATT&CK. Authors in [79] presented a system that automatically chooses cyber deception strategies based on the information it obtains from unstructured CTI reports through NLP. It first maps the extracted Indicators of Compromise (IoCs) to the relevant MITRE ATT&CK tactics and techniques, which then allows for the prediction of the attacker's actions. The deception actions proposed by this framework are in line with the MITRE D3FEND [80] and MITRE Engage [81] frameworks, which are based on the aforementioned profiling.

By aligning deception strategies with specific ATT&CK techniques, defenders can deploy assets that disrupt adversary decision-making, enhance situational awareness, and collect high-fidelity threat intelligence. Fig. 3 presents an example of the Enterprise ATT&CK matrix techniques, annotated with corresponding deception technique, as illustrated by MITRE D3FEND, and also mapped to the corresponding engagement techniques, as described by MITRE ENGAGE.

V. Intelligent Orchestration of Cyber Deception through Reinforcement Learning and Game Theory

The Cyber Kill Chain and MITRE ATT&CK frameworks serve as a guide for the mapping of adversary behaviors to suitable deception techniques, but the actual deception deployment in dynamic environments would need continuous adaptation and decision-making under uncertainty. In this way, reinforcement learning (RL) and game theory (GT) become central for the intelligent orchestration of cyber deception. Their embedding into the threat modeling process not only helps in contextualizing but also promotes the continuous improvement of deception strategies, thus providing smart and proactive defense across a wide range of areas.

Game-theoretic models capture the strategic interactions between attackers and defenders, allowing defenders to find the optimal countermeasures by taking the adversarial payoffs, costs, and bounded rationality into account. Authors in [82] introduced a game-theoretic framework to optimize the deployment of honeypots and software diversity in a network under resource constraints. It presents two-layer deception tactics: the first layer determines the position of honeypots on network edges using a zero-sum game based on attack graph analysis and node importance, while the second layer employs a non-zero-sum game to manage software diversity across honeypots, thereby preventing the attackers from recognizing the uniform deception. The model takes into consideration the attacker-defender interactions and provides the mixed-strategy Nash equilibria to attain the maximum defender payoff. The experimental results indicate that the strategic allocation of honeypots with deception diversity significantly enhances the defense effectiveness against adaptive adversaries.

In [24], the authors proposed a novel game-theoretic framework for honeypot allocation in dynamic tactical networks. It makes use of the fact that node mobility changes network topology and attack paths. The interaction modeled herein is between the defender and the attacker in a two-player zero-sum Markov game on changing attack graphs. The defender is the one who wants to place honeypots in a way that the attacker's deception is maximized and the costs for reconfiguration in the network states are minimized, while the attacker is the one who picks the right attack paths. A predictive model has been incorporated and a solution has been provided for stationary Nash equilibria through the application of a Q-minimax algorithm. The results of the investigators show that projecting the future mobility of the system would be the most effective defense measure, while it also reduces the rewards of the attacker's efforts. Thus, it becomes a method that is both scalable and adaptive for the plan of deception in a mobile and resource-constrained environment.

In [83], the authors provided a foundational game-theoretic framework for applying cyber deception through evidence-based signaling games. They provide mathematical models that capture the dynamics of strategy between a deceiver and a deceived under uncertainty, including the probabilistic evidence of intrusion detection outputs. Also, they analyze: a binary state deception, a continuous state deception with a detection cost, and a multi-level deceptive modeling of APTs. Perfect Bayesian Nash Equilibrium (PBNE) is used to illustrate the optimal deception strategies. This research introduces the term of deceivability in a formal way, and states certain conditions for the deception to be not only sustainable but also effective. It supports the design of adaptive, long-term deception policies by considering how attackers learn and update beliefs over time, making it a theoretical cornerstone for strategic deception planning in adversarial settings.

In [84], the authors proposed a game-theoretic framework

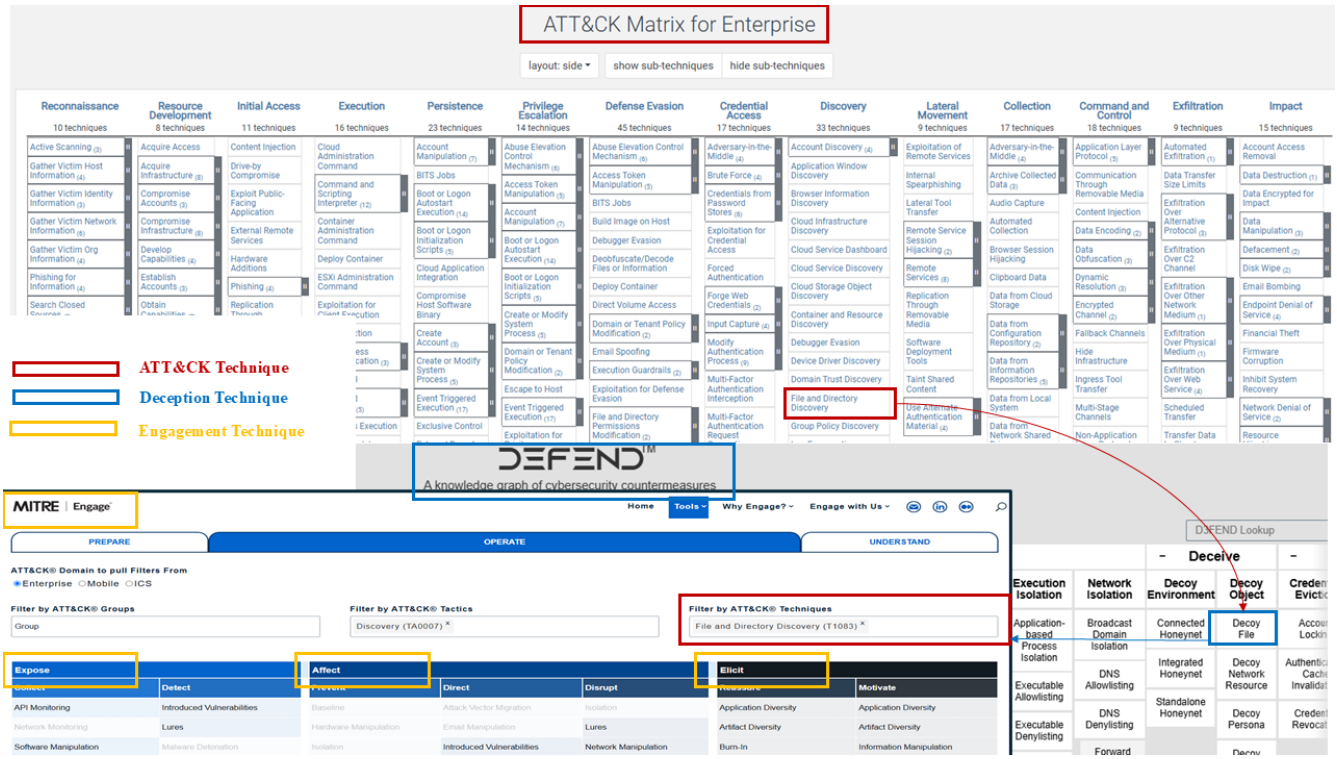

Fig. 3. Mapping of an enterprise MITRE ATT&CK technique to corresponding deception and engagement techniques.

that has multiple levels as well as learning and control theories. The main application is to counter deceptive information attacks in Intelligent Transportation Systems (ITS). The introduction of the PRADA (Proactive Risk Assessment and Mitigation of Misinformed Demand Attack) framework, which regards the attacker-defender dynamics as a stackelberg game, and quantifies both local and network-wide effects of demand manipulation, is the most important contribution. Besides, the paper also proposes trust-constrained recommendations as a tool to ensure system reliability in case of attack. The approach integrates strategic modeling, user equilibrium routing, and predictive resilience mechanisms, where game theory emerges as the core instrument in facilitating adaptive deception in the ITS environments.

On the other hand, reinforcement learning facilitates data-oriented, self-adaptive decision-making by obtaining deception policies over time through trial-and-error interaction with simulated environments. The techniques, such as Deep RL and Q-learning provide defenders with the opportunity to dynamically modify the placement of the deception, the interaction fidelity, and the signaling strategies according to the new threats. Authors in [85] proposed an adaptive Q-learning-based framework for orchestrating cyber denial and deception strategies in enterprise networks, by using a detailed attacker-aware attack graph. The model integrates MITRE ATT&CK tactics and CVSS-derived metrics to evaluate vulnerabilities in terms of importance, exploitability, and damage. Two different attacker profiles: opportunistic and strategic, are used to model different behaviors, and the defender learns optimal responses (none, denial, deception) via Q-learning. The approach enables a dynamic and tailored honeypatch placement that is built to the attacker's strategy and system context. Besides, to enhance decision making, some works combine reinforcement learning with game theory that enables defenders to learn from gained experience and think strategically about the attacker's behavior.

In [86], the authors suggested a reinforcement learning approach to dynamically select and orchestrate deception and Moving Target Defense (MTD) strategies. They developed a confrontation model, which is a prototype to simulate the interaction between an attacker and a defender, taking into account perception asymmetry and phase progression via a simplified Cyber Kill Chain (CKC). They employ Deep Q-Learning to train a defensive agent capable of adapting its strategy based on observed attacker actions and system states. The major contribution is showing how deep reinforcement learning can enable adaptive, perception-aware, and cost-sensitive orchestrations of deception and MTD in complex environments, which are better than static or naive strategies in all tested scenarios.

In [87], the authors developed a deep reinforcement learning framework, a novel solution for the deployment of honeypot strategies in complex networks represented using Bayesian attack graphs. They model a stackelberg game that involves a defender (the one who deploys a honeypot) and a hacker, where the defender takes the lead by deploying high- or low-interaction honeypots to manipulate the hacker. The setting is represented as a Markov Decision Process (MDP), and the search for the strategy space is carried out using Actor-Critic and Proximal Policy Optimization (PPO) algorithms. Their method contributes to the dynamic, state-aware decision making and is reported to outperform random and value-weighted baseline strategies significantly in experiments. The study presents the effectiveness of RL in learning robust and

cost-efficient deception strategies under uncertainty.

In [88], the authors proposed a hybrid game-theoretic and reinforcement learning method to enable quantum superdense coding security against advanced attacks, including scrambling and bijection. The model introduced decoy qubits as deception elements so that eavesdropping attempts can be detected, and the interaction between the sender/receiver (Alice and Bob) and the attacker (Eve) is framed as a non-cooperative game. The authors used custom utility functions: Return on Protection and Return on Attack, analyzed both pure and mixed strategy Nash equilibrium, and used a Q-learning approach to optimize decoy insertion strategies under uncertainty. The simulation results confirm the model's capability to converge to equilibrium and minimize the false negatives, which illustrates the efficiency of game-theoretic reasoning coupled with RL-driven strategy adaptation in quantum security.

VI. Cyber Deception in Cloud Environments

Cloud computing has brought a drastic change in how organizations can deploy and scale IT infrastructure, providing elasticity, multi-tenancy, and on-demand resource provisioning. However, many of these are the same design features that create unique security problems, such as co-residency attacks, lateral movement, insecure APIs, and dynamic reconnaissance. The dynamic, decentralized, and virtualized characteristics of the cloud infrastructure make it an attractive target for attackers, while legacy-based perimeter security approaches frequently do not secure the evolving threats adequately [89]. Deception has been the proactive cyber defense technique to address these threats with the help of misleading artifacts, such as honeypots, decoy virtual machines, fake credentials, and deceptive storage buckets. Novel methodologies are being employed nowadays, such as IP and VM shuffling, AI-driven decoy placement, and adaptive honeynet orchestration for the purpose of confusing respective attackers, postponing their advancement, while collecting useful threat information. Installation of the deception in cloud environments is not only through this promise but also through the involved challenges, such as ensuring the scalability and realism of deceptive elements, avoiding interference with legitimate operations, and integrating virtualized orchestration layers in a seamless manner. This section covers cyber deception in the cloud setting, particularly focusing on deception-aware architectures, the corresponding cloud (IaaS, PaaS, SaaS) threat coverage, and the limitations and deployment trade-offs that current methods come with.

In [90], the authors proposed an integrated cloud security framework based on a mixture of signature-based NIDS (SNORT) with distributed honeypot networks (Glastopf, Cowrie, Dionaea) in an OpenStack environment. The architecture can be seen in Fig. 4, where authors have strategically placed SNORT at network controller nodes (to deal with external traffic) and compute nodes with OpenV Switches (for internal/local traffic), while honeypots emulate services in order to capture interactions of the attacker. The data obtained is then analyzed in a sandbox environment (Cuckoo Sandbox) to dynamically extract behaviors of the malware, which then triggers an automated use case to add new and update the current SNORT rules. The system shows an increase of unknown attacks detected (for example, XSS, malware variants) as well as causing redundant alerts, among which, we should mention the overhead of distributed NIDS instances and that the research is focused mainly on network-layer attacks, but the added value of integrating host-based defenses also like VM introspection is still valid.

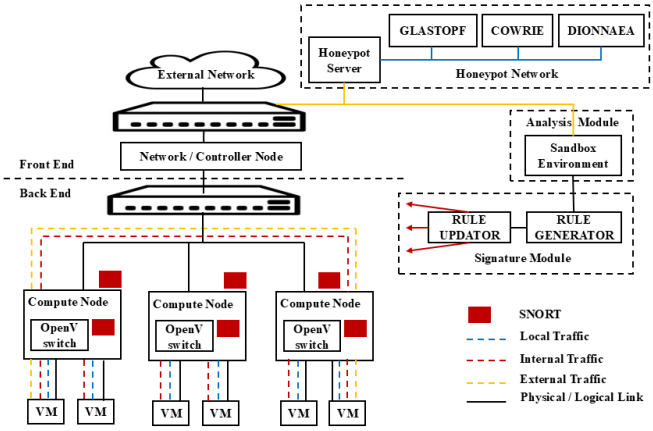

Fig. 4. The overall architecture showing the different modules and types of traffic in [90].

In [91], the authors provided a game-theoretic Moving Target Defense (MTD) framework that tactically combines both real and deceptive virtual machine (VM) migrations to ensure security of the cloud environments against side-channel and co-residency attacks. They proposed a new model that represents the security interaction as a signaling game, in which defenders can either perform live/non-live migrations or simulate them by carefully crafted traffic patterns, creating uncertainty for the attackers while minimizing costly real migrations. The analysis shows that Nash equilibria are the only beyond which deceptive signaling offers the same security level as physical migrations, the network controller operating on detection alerts and cost parameters that dynamically set best strategies. This study of the symbolic trade-off of migration overhead and security gains, thus admitting both a theoretical model for the deceptive MTD in clouds and a practicable decision algorithm for the implementation, clearly underlines its contribution, whereas questions regarding performance in realistic situations and adversarial learning remain unaddressed.

In [71], the authors suggested the Automated Honeynet Deployment Strategy (AHDS), which is a dynamic security framework for container-based clouds that adaptively deploys honeypots to prevent emerging threats. AHDS uses an Attack Graph (AG) to chart hypothetical routes of attacks, and it also comes up with a Deceptive Exploring Surface (DES), a network of strategically placed source honeypots to divert potential enemies from critical assets. The algorithm gives priority to high-value nodes based on the degree of centrality and optimizes the layout with the help of a set cover-based algorithm, which has the effect of not wasting resources and having the maximum possible coverage. The system is built upon three main building blocks, which are: 1) a Monitor Engine for recording the infrastructure changes and costs, 2) a Decision Engine that adapts the strategies of the honeynet in real time based on the AG assessment, and 3) a Deployment Handler for automatic changes of configurations. Evaluations

reveal that AHDS is 83% more efficient at decreasing the possible attack emergence while outgunning the traditional strategies (e.g., "Max" and "Most") both in effectiveness and in no resource waste, and is seamlessly adopted by datacenter sizes. Nonetheless, its effectiveness hinges on precise AG modeling and it relies on assumptions that intruders follow predictable propagation patterns, potentially limiting robustness against novel or adaptive threats.

In [92], the authors suggested a cloud-based deception framework that can disrupt DDoS reconnaissance by using SDN/NFV technology to generate virtual reflection networks in the cloud, which send back to the attackers false topology data but still keep correct details for the legitimate users. The system is using GRE tunnels for the rerouting of the reconnaissance traffic (traceroute/ping) to the virtual SDN switches, which are programs that implement IP hopping and path mutation, with the physical and cloud networks being synchronized by dual SDN controllers and NFV orchestrator. The solution was tested on the GENI platform and only added a 25-35ms delay, decreased packet loss to below 1%, and reflected network alterations in 75ms, while it was >99% cheaper than a physical infrastructure, but it is reliant on the accuracy of the synchronization that is required to connect between the physical and the virtual networks and also it can be difficult in the situation of very frequent or large-scale topology changes.

In [93], the authors proposed a deception-based defense framework for private cloud infrastructures that is based on using honeypots in an enterprise IT environment to identify and contain internal threats by deploying decoy virtual machines (VMs) with emulated services (e.g., routers, databases) that run parallel to the production VMs. The setup, as illustrated in Fig. 5, includes Snort for intrusion detection and Sebek for activity logging, which together log attacker interactions and pass the data to a Multiclass Support Vector Machine (MSVM) classifier that applies feature selection (Gini index) and dimensionality reduction (PCA) methods to traffic analysis. With the help of KDDCup99, NSL-KDD, and Gure-KDD datasets, the classifier achieves a remarkable 95% accuracy in distinguishing between different types of attack (DoS, U2R, R2L, probing) and normal behavior by thus effectively isolating production systems and profiling intruder activity during the process. The application, while functional against most network-layer threats, is limited by the framework's signature-based detection approach that is inadequate for dealing with advanced attacks, which suggests improvement through infrastructure integration with anomaly detection and threat intelligence feeds for a more comprehensive security coverage.

In [94], the authors introduced a novel AI-based defensive deception framework to secure multi-tenant cloud environments from reconnaissance attacks. The strategy overcomes the issues brought about by static, uniform configurations in cloud networks, which add simplicity to reconnaissance for the adversaries. To this end, the framework exploits deep reinforcement learning (DRL) to formulate the optimal deployment of decoys. A utility function models common OS vulnerabilities as a threat, while a DRL agent who has been trained with Proximal Policy Optimization (PPO) is the one to produce strategies for configuring and distributing decoy virtual machines in a fine-grained manner. These strategies aim to hide actual assets

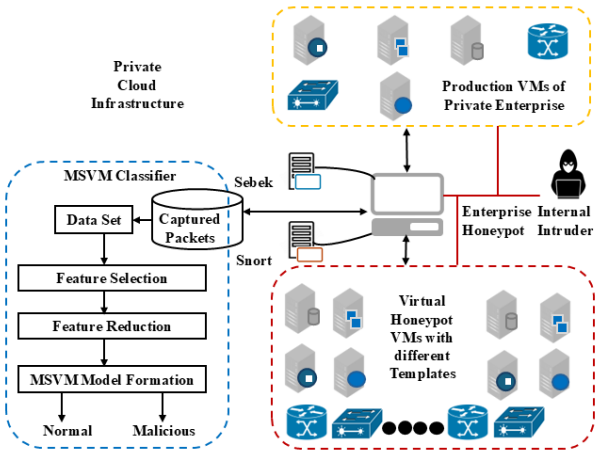

Fig. 5. The private cloud infrastructure with enterprise honeypot in [93].

and thereby increase the attacker's resource consumption. Conducted simulations proved the framework's skill in improving asset concealment by 20.58% and raising attack costs by 40.4% as compared to the traditional approach. This work challenges the idea that DRL constitutes the only probability of deception being dynamic and scalable in the clouds, thus allowing for accurate protection against reconnaissance.

In [95], the authors proposed the use of IRDS4C, a multilayer deception framework for the aim of detecting zero-day ransomware and also intrusions across cloud environments such as IaaS, PaaS, and SaaS. Through deploying of strategically positioned decoys, including high-interaction honey files/tokens (to mislead the attackers) and low-interaction canary resources (for early detection), the system can use the ASCII-based naming scheme to make the ransomware interact with the decoys first. The behavioral monitoring that it uses to track unauthorized access, does not however, disturb the legitimate operations. The testing was performed on Google Cloud with seven ransomware families (e.g., REvil, WannaCry) and IRDS4C was found to achieve 100% detection accuracy, while faster than traditional methods like file hashing. Besides that, the framework also successfully recognized the human intruders in simulations of different breaches carried out, being a lightweight, scalable solution designed for cloud-native deployment, with future extensions planned for Linux and macOS, compatibility.

In [96], the authors introduced a novel containerized honeypot-based deception system which is capable of real-time tracking and profiling of cyber adversaries in the cloud setting. Docker containers are the framework's key feature that are used to establish a variety of honeypots (e.g., Cowrie, Dionaea, DDoSPot, RedisHoneypot) in different Microsoft Azure regions, under the support of ELK stack (Elasticsearch, Logstash, Kibana), which will be used for data aggregation and visualization. The system looks like a typical cloud service and to create open ports for the attackers to connect to it uses a Suricata, a tool for p0f, and FATT, along with some interesting extras whose CVEs, operating systems, and behavioral patterns are being analyzed. The deployment of the work was successful for the reason that it caught a large number of attacks, such as the exploitation of the old CVEs,

the botnet-based DDoS traffic, and the suspected URL usage, which provided intelligence on the adversary techniques and the infrastructures. The experiment setup has shown competitiveness in processing data acquisition, forensics, and attack identification, with plans for the next stage to include the early-warning system upgrade and the threat coverage expansion to the zero-day detection. This work is a representative example showing how scalable and cloud-native honeypot deployments can act as a strategic deception tool for proactive cyber defense.

In [97], the authors suggested a resource-aware cyber deception framework for microservice-based cloud-native applications. This is another option in the case of traditional pre-built decoys that are not effective in dynamic service environments because of their inability to blend. Their design, using the distribution of a non-linear mixed integer programming problem, employs the production microservices cloned as high-fidelity decoys to catch lateral movements of the attacker that are being made along the attack path, at the same time maximizing the intercepted attack paths with respect to CPU/RAM restrictions and also considering the attack graph topology. To solve the problems related to computational complexity, they have offered an approach that enables the prioritization of the most critical microservices based on the betweenness centrality measure. The algorithm provided is highly scalable, and the realistic placement of the decoys is achieved. Their results clearly show that, even with fewer resources, the coverage gained from deceptive means was more than with the basic approach; thus, the framework can be confidently implemented in modern cloud spaces where contextual and resource-efficient deception are a must.

Table II summarizes all the discussed works in cloud environments, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

VII. Cyber Deception in Wireless Networks

Wireless networks are largely under threat due to the inherent challenges posed by their broadcast nature, limited resources, and dynamic topologies, which make them susceptible to a range of cyber-attacks including reconnaissance, eavesdropping, jamming, spoofing, and lateral movement attacks. Security mechanisms that are based on encryption, authentication, and intrusion detection are the most commonly used traditional safety measures. They are not enough, in decentralized networks, as open networks are, with their limitations in computational power and energy efficiency of many wireless devices [98]. In response, this has been countered by the emergence of cyber deception, where this countermeasure aims to create trouble for attackers, ensure the communication of critical information, and collect data on potential threats. For instance, the defenders do not just deploy decoy nodes, fake access points, adaptive honeypots, but also the communication protocols that mislead the attacker. Such practices take the place of the real nodes and thus prevent the attacker from gaining the proper situational awareness. Recent works focused on the recent developments, especially in software-defined and virtualized wireless networks, where the use of AI and machine learning has been instrumental in supporting the dynamic traffic redirection, strategic decoy deployment, and behavior modeling of the attacker. Despite its large potential, the operationalization of deception in wireless networks also encounters issues like keeping the decoys realistic, reducing the interference with the real communication, and ensuring rapid adaptation to network change. This section reviews the most important works in the field that utilize deception as an adaptation mechanism to the constraints and shifting hostile environment in wireless communication systems.

A trilogy of deception-driven architectures is introduced to secure the virtualized wireless networks against both RF-based and infrastructure-layer threats. In the first work [99], the authors introduced a cyber deception framework for virtualized wireless networks that counters unauthorized access and DoS attacks through SDN-controlled dynamic redirection. An SDN controller observes the network traffic and redirects adversarial flows to a Deception mobile virtual network operators (DMVNO), which acts as a decoy through the emulation of legitimate services, while the availability for benign users is preserved. The defense mechanism characterizes the attack dynamics as a birth-death process and assesses them using Monte Carlo simulations with metrics such as attack redirection rate, deployment timeliness, and the probability of legitimate users being blocked. Primary limitations are noted, including the failure to inject real-time adaptability for unidentified attack patterns and the issue related to the delays in the detection-to-deployment pipelines not being mentioned. These limitations are suggested as future work. To address this, the second paper [100] introduced Deceptor-in-the-Middle (DitM), as illustrated in Fig. 6, which is a cyber-deception architecture that can be found in virtualized wireless networks and provides concurrent real-time attacker detection and dynamic adversarial redirection. At the beginning of the process, the system detects adversaries by utilizing RF energy sensing and Wald's Sequential Probability Ratio Test (SPRT), then, it transparently routes them to a Deception VNO (D-VNO) through ARP cache poisoning, by mimicking a Man-in-the-Middle (MitM) attack in the opposite way. The framework is built using an SDN controller and network aggregator that operate in a "sense-observe-manipulate" mode, with the added benefit of ensuring that genuine users do not experience any service disruption. The distribution reflects a comparison between the detection rates (false alarm and miss-detection rates) at different thresholds, with the only limitation being that it lacks formal guidelines for the optimal resource utilization and leaves scalability, as well as, adaptive adversary management to future work. The third paper [101] addressed the virtualized wireless networks cyber deception optimization using a Stackelberg game-theoretic model, in which an SDN controller, under budget constraints, strategically maps actual RF configurations to deceptive observable configurations while interacting with either naive or rational attackers. The work demonstrates the NP-hardness of the optimal strategy selection and presents a polynomial-time algorithm for naive attackers as well as a scalable greedy heuristic (GMS) for rational ones, proving successful attacker utility minimization and computationally efficient deception deployment through simulations. While the steps are made from static to adaptive and after that, to game-theoretically optimized deception, the method is still limited by its assumptions about rational attacker behavior and also faces challenges with optimal Mixed Integer Linear Programming (MILP) solutions on large networks that have scalability issues, thereby leaving dynamic configuration adaptation and real-

TABLE II. SUMMARY OF CYBER DECEPTION TECHNIQUES IN CLOUD ENVIRONMENTS

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[90]	Mitigate known and unknown attacks in cloud environments	SQL Injection, XSS, SSH attacks, Malware injection	High/low interaction honeypots	NIDS Module (SNORT), honeypot network module (Glastopf, Dionaea, Cowrie), analysis module (Cuckoo Sandbox), signature module (Rule Generator and Updater)	Reduction in generated alerts, detection of unknown attacks via custom rules, dynamic malware analysis results	Overhead due to multiple NIDS instances not studied, focus limited to network-based attacks, no integration with HIDS, ACLs, or firewalls, performance-security trade-off not quantified
[91]	Mitigate cloud-side attacks (e.g., side-channel) while minimizing migration costs	Side-channel attacks (intra/inter-host), VM co-residency threats	Signaling game-based deceptive VM migration (live/non-live or fake signals)	Network Controller (alerts and migration decisions), VMs (defenders in the game), attacker (monitors VM signals), signaling Module (generates live/non-live migration or deceptive signals)	Cost savings, security gain, equilibrium stability	Focuses on VM-level attacks, ignores host/network-layer threats, no real-world validation (simulation only), unquantified impact of deception on performance, requires attacker belief modeling (p,q), which may not hold in practice
[71]	Mitigate dynamic threats in container-based clouds	Multi-stage lateral movement	Automated honeypot placement using Attack Graph and Deceptive Exploring Surface	Monitor engine (tracks system changes), decision engine (optimizes honeypot placement), deployment handler (configures/ deploys honeypots), honeypot nodes	Attack success rate reduction, resource efficiency, scalability, flexibility	Depends on attack graph accuracy, static assumptions on attacker model
[92]	Disrupt reconnaissance for DDoS attacks (e.g., Crossfire, Coremelt) without affecting users	Reconnaissance phase of DDoS	Cloud-based reflection network: GRE tunnels forward probes to virtual topology, SDN-enabled IP hopping for fake paths, "Reflected" virtual topology mimics physical network	Physical SDN switches, cloud virtual switches, local/Cloud SDN controllers, NFV management host	response time, packet loss, update latency, cost	Requires tight sync between physical and virtual topologies
[93]	Divert and detect internal threats in private clouds	Internal reconnaissance, probe, DoS, User to Root (U2R), Remote to Local (R2L)	Enterprise honeypot with VM-level emulation	Honeypot VMs, Snort, Sebek, MSVM classifier	Classification accuracy, alert generation	Limited to internal threats, dependence on known attack patterns
[94]	Proactively counter reconnaissance attacks in cloud environments	Network scanning, probing, OS fingerprinting, exploitation of common OS vulnerabilities	DRL-based decoy VM placement	DRL agent (PPO), OS vulnerability model, simulated tenants	Deception entropy, attack cost, Joint-Defense Goal	Limited to OS-layer vulnerabilities, scalability
[95]	Detect zero-day ransomware attacks and intruders in cloud environments (IaaS, PaaS, SaaS)	Ransomware (e.g., REvil, WannaCry, NotPetya) and intrusion attempts (e.g., unauthorized access to cloud storage/ servers)	High-interaction decoys (fake files/tokens), low-interaction decoys (canary files/tokens), ANSI/ASCII-named decoy files	Decoy resources (files, tokens, partitions, folders, servers), event handler watcher (monitors interactions with decoys via cloud API hypervisor)	Detection rate, detection time, speed vs. hashing/entropy methods	Lacks compatibility with Linux/MacOS, focus on Windows-based cloud, file system reliance (requires NTFS for positioning technique)
[96]	Monitor and attribute cloud attacks in real time	Botnets, CVE exploits, DDoS	Containerized honeypots, high-interaction (e.g., Cowrie SSH/ Telnet), low-interaction (e.g., Dionaea, ADBHoney) decoys	Azure cloud, ELK stack, Suricata, P0f, FATT, honeypots	Real-time logging, CVE capture, attacker attribution	Manual signature extension, needs zero-day attack adaptation
[97]	Strengthen security in cloud-native microservice architectures by intercepting and mitigating malicious lateral movements of attackers	Lateral movement, container compromise	High-fidelity decoys (clones of legitimate microservices)	Microservice architecture, decoy containers	Computation time, decoy interaction probability, average decoys per attack path	Assumes attack graph availability, computational complexity, node constraints

world adversarial creativity as open challenges for future work. This illustrates a transition from simple redirection to strategic deception in wireless virtualization security.

In [102], the authors suggested counteracting the jamming (which means that an adversary injects noise to decrease of signal-to-noise ratio (SNR) and disrupt communication) attacks on wireless networks using game theory. Their approach concerns a dual channel system, which has a real and a fake transmitter, and a receiver pair, where fake transmissions mislead the jammer into splitting its power of jamming to both channels. The interaction is formulated by a Stackelberg game, in which the system (the leader) performs the optimization of power allocation first, and the jammer (the follower) afterwards senses and reacts to the channel activity. There are two cases discussed: 1) a non-strategic jammer with fixed behavior,

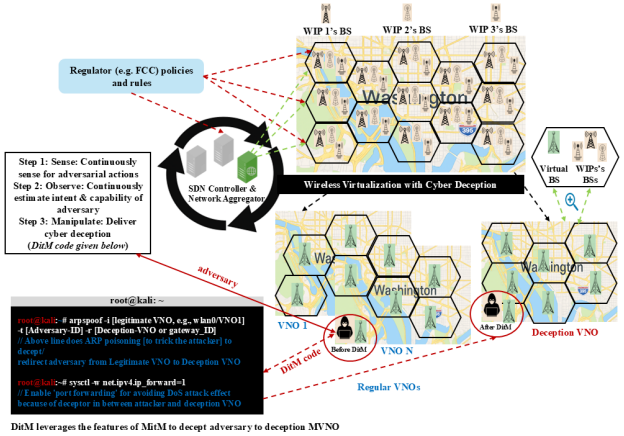

Fig. 6. An example scenario for cyber deception in wireless virtualization in [100]: logical redirection of adversary from VNO N to deception VNO via subleased base station.

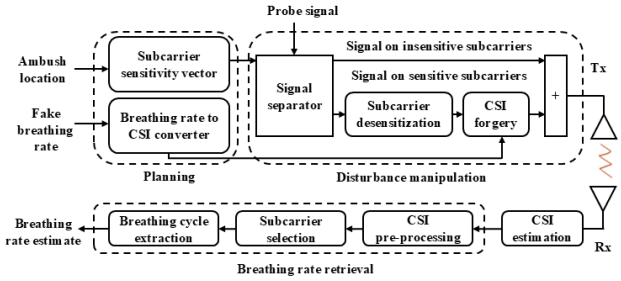

Fig. 7. The flow chart of the proposed HoneyBreath defense in [103].

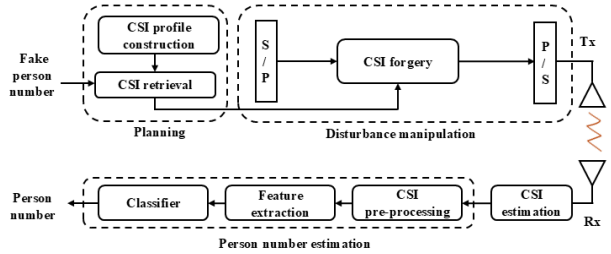

Fig. 8. The flow chart of the proposed Ghost defense in [103].

and 2) a strategic jammer, for which they can establish a Subgame Perfect Nash Equilibrium (SPNE) for optimal power allocation. The simulations validate that the technique reduces the jammed signal and boosts the legitimate channel's transmission rate. However, the work presuming the total efficacy in jammer sensing is limited to just two orthogonal channels and also does not deal with involving scalability in multi-node or dynamic environments. Despite these shortcomings, the paper underscores the fact that power-based deception is a way of significantly increasing resistance to reactive jamming even in simple wireless systems.

In [103], the authors presented two deception-based techniques, depicted in Fig. 7 and Fig. 8 that are used to withstand wireless privacy threats such as the Channel State Information (CSI) that are used by the attacker. HoneyBreath is a method that gives the wrong breathing rate by first hiding the sensitive subcarriers and then sending forged sinusoidal CSI patterns, while Ghost controls all the subcarriers by using previously recorded CSI to fool crowd counting classifiers. These strategies have been applied on USRP X310 platforms and they have given good evaluation results. HoneyBreath has reached the state at which the errors of breathing rate estimation go below 1.2 bpm; on the contrary, Ghost has had complete occupancy detection success. However, they require attackers' location beforehand and pre-collected CSI profiles, which might hinder their use in some dynamic cases. The study illustrates that strategic CSI manipulation could be a viable method for achieving privacy against wireless inference attacks while being largely unnoticed and practical.

A two-part contribution to deception-based defense strategies in wireless sensor networks (WSNs), specifically targeting energy depletion and battery drain denial-of-service (DoS) attacks is presented. The first study [104] presented a deception framework designed to assist cluster heads (CHs) in wireless sensor networks (WSNs) against energy depletion attacks using a game-theoretic approach. The strategy involves both high- and low-interaction honeypots, which are placed in a secure zone, governed by an intrusion detection system (IDS), that counts attackers (sophisticated/ non-sophisticated) and directs them to suitable honeypots. An incomplete information Bayesian game model derives Nash equilibrium strategies for CH energy preservation and attack mitigation. In MATLAB, simulations were carried out, and the approach was validated, which showed a decrease in attack success rates through the reallocating of honeypots dynamically. Major constraints include a static attacker assumption and a sole reliance on predefined honeypot configurations, thus demonstrating the scalability and real-time adaptability issues for the next work. The second study [105] builds on this framework by proposing a lightweight defense mechanism to battery-draining DoS attacks in wireless sensor networks (WSNs) through a signaling game framework where sensor nodes smartly release deceptive energy-level signals in order to misdirect attackers. The authors depicted the relationship among nodes and attackers as a two-player Bayesian game to derive the optimal defense strategies through Perfect Bayesian Nash Equilibrium analysis, showing that the nodes can effectively save energy by trying to conceal the true state when the costs of lying are low. The simulations show that by using the deceptive signaling, the attacker's utility is reduced while the nodes are being protected. However, the approach only considers single-round interactions and does not include coordinated attacks or network failures that could be implemented in future work on dynamic game models and real-world implementation in platforms such as NS3 or Cooja.

Table III summarizes all the discussed works in wireless networks, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

VIII. Cyber Deception in Cyber-physical System

Cyber-Physical Systems (CPS) are systems that consist of computational (cyber) elements and physical processes and are done almost as one, which enables real-time monitoring, control, and interaction with the physical world. These systems are virtually the key to numerous essential fields

TABLE III. SUMMARY OF CYBER DECEPTION TECHNIQUES IN WIRELESS NETWORKS

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[99]	Isolate adversarial users in virtualized wireless networks	Unauthorized access, DoS, dynamic attacks	Deception MVNO (DMVNO): a decoy virtual network operator	SDN controller, Wireless Infrastructure Providers (WIPs), legitimate MVNOs, deception MVNO	Blocking probability, attacker arrival rate, deception deployment time	Detection delay (time gap between attack detection and DMVNO deployment)
[100]	Protect virtualized wireless networks from adversarial attacks while maintaining QoS for legitimate users	Impersonation attacks, location falsification attacks	Deceptor-in-the-Middle (DitM): SDN-controlled redirection (legitimate VNOs act as "middle-men" to reroute adversaries to deception VNOs)	SDN controller, wireless infrastructure providers (WIPs), legitimate VNOs, deception VNOs	Detection range, false alarm rate, throughput	Lacks optimal strategy selection mechanism
[101]	Secure virtualized wireless networks by deceiving attackers into interacting with decoy systems	Jamming/DoS attacks, reconnaissance attacks	Stackelberg game-based mapping of true to fake RF bands	SDN controller, wireless infrastructure providers (WIPs), VNO, deceptor VNO	Attacker utility, deception cost, feasibility constraints	Scalability, computational complexity
[102]	Mitigate jamming attacks to protect the transmission of real information between a transmitter-receiver pair	Jamming attacks	Fake signal transmission on orthogonal channel	Twotransmitter-receiver pairs, orthogonal channels, jammer	Transmission rate, optimal power allocation, utility equilibrium	Assumes perfect sensing, static two-channel setup
[103]	Protect user privacy by preventing accurate inference of breathing rates and crowd counts via wireless signals	Channel state information (CSI) wireless inference attacks	HoneyBreath (transmits fake CSI to deceive eavesdroppers into inferring incorrect breathing rates), Ghost (manipulates CSI to fabricate false crowd counts in empty rooms)	Software-defined radio (USRP X310) platforms, CSI profile library, multiple antennas	Breathing rate error, crowd counting	Static or predictable attacker positions, pre-collected CSI profiles
[104]	Protect cluster heads (CHs) in Wireless Sensor Networks (WSNs) from energy depletion attacks	Energy depletion attacks	High- and low-interaction honeypots mimic CHs to lure attackers	Cluster-based WSN with CHs and member nodes (CMs), secure zone with IDS, honeypots	Nash equilibrium analysis, probability of attack strategies, probability of defender strategies	Static attacker beliefs, single-round interaction
[105]	Mitigate battery-draining DoS attacks in WSNs	Battery drain DoS attacks	Bayesian signaling game with deceptive energy state broadcasting	Regular node (sender), attacker (receiver), signaling framework	Attack probability, PBNE types (pooling/separating)	Does not address cooperative attackers or network failures

from self-driving vehicles to energy and health care, factory robots, and industrial automation. A typical CPS is structured in the following form: sensors and actuators interface the system with the physical environment, embedded controllers are responsible for executing control algorithms, networked communication is in charge of data exchange; cloud or edge infrastructure are resources for computation and analytics. More CPS, in general, are more vulnerable to attacks as they are focusing on wireless communications, open protocols, and internet connectivity. This entails the introduction of such vulnerabilities as time-line constraints, limited computational resources, insecure firmware, and insufficient authentication. Threats specific to CPS can be the origin of physical damage, privacy issues, or huge business interruption. Some of the most notorious cases, like remote hijacking of vehicles or manipulation of smart grid components, are a lesson on how traditional security practices can't protect CPS satisfactorily by themselves [106]. Cyber deception finds a place here as a likely solution by putting the attacker off track with signal, data, node misdirection, or wrong behavior models that divert the attacker's perception and planning. This Disinformation can slow down enemies, divulge intelligence operations, and aid in adjustments to threats. However, the usage of untruth in CPS is complicated as the systems' timely response and safety-criticality requirements make it hard, and they need to be taken into account during the design phase. Moreover, the fact that adversaries are sophisticated enough to detect badly implemented disinformation is an additional inconvenience. This section presents cyber deception methods contextualized to CPS environments with an emphasis on deception-aware system design, attacker engagement strategies, and domain-specific constraints.

In [107], the authors suggested the use of a deception-as-defense framework for CPS, in which strategic misinformation is utilized to change the perception of rational adversaries who are trying to estimate the states of the system. The interaction is modeled as a stackelberg game, in which the defender (leader) first of all commits to a signaling policy while the adversary (follower) reacts optimally. The paper shows the results under Gaussian-distributed system states and quadratic cost functions that linear signaling rules (which include optional noise) are the ideal means to indirect adversarial behavior without direct enforcement. The solution, obtained through semi-definite programming (SDP), is also valid for dynamic situations, partial observations, and unknown attacker goals, and it is robust under both Bayesian and non-Bayesian uncertainties. The study provides a foundational game-theoretic approach in protecting CPS roads against adversaries, focusing on the indirect channel perceptual control, which is a key for resilient systems that require both stealth and accuracy.

In [108], the authors introduced a honeypot-based decep-

tion framework for CPS that optimizes the defensive strategies model under limited human analysis resources by modeling attacker-defender interactions, which are represented as a Bayesian game with missing knowledge. The proposed method involves the use of both low- and high-interaction honeypots integrated into the CPS cyber layer, and using the derived optimal configurations for defense and resource allocation to ensure the highest defensive payoff, proving the inclusion of several Bayesian-Nash equilibria under different analysis budgets. Through the game-theoretic approach, they clarify the distinction between the types of attackers (weak vs. strong offensive access) and defense mechanisms (service provision vs. non-provision), thanks to the simulations they carried out, which revealed that turning attention to the honeypots that provide a greater unit analysis utility increases the efficiency of the capturing process. Up to now, the framework has improved the performance of practical deception in CPS, though it is believed that the strategies of attackers should remain static and interactions are to be single-stage, it has put off the adaptation to dynamic and multi-stage attacks for the upcoming tasks.

In [109], the authors presented a honeypot-based intrusion detection system, specifically for CPS, which addresses threats such as spoofing, code injection, malware, and DoS that can disrupt the ordinary physical operations. The architecture they used diverts all the traffic into a virtual or hybrid honeypot that is placed in a strategic location and exploits the high availability of real-time attack detection, logging, and blocking, thus achieving a cost-effective and scalable deployment (e.g., Cisco, Conpot). The work makes a comparison of the interaction level, scalability, and resource requirements of the various honeypot types, which is a good point to present them as the main instruments for the early threat detection of CPS systems, given the fact that conventional methods often fail in real-time conditions. Nevertheless, the present version is applied to single systems only, and there is a need for multi-device testbed with dynamic traffic patterns, which would be the practical deception-based approach for the security of CPS.

In [110], the authors presented a resilient and smart technique to identify the honeypots in softwarized industrial CPS that largely depend on virtualization and network function slicing. Its main architecture combines a secure fuzzy testing approach with a deep learning-based classification model to address the problem of recognizing real devices from honeypots in heavily virtualized network environments. The first module of the architecture, as illustrated in Fig. 9, creates probe packets using multi-object mutation strategies based on security rules, which guarantee that probe queries are safe but rich in features that are not handled correctly between real devices and honeypots. These are the optimized packets that will be used to scan online devices. The additional responses that are returned are then analyzed for features that are found only in the honeypots and are classified by a CNN model that has been trained before. The overall design provides the packet creation and identification phases separately in order to enhance the scalability and minimize risks. Tests carried out demonstrate that, in addition to being fast and safe, the method has outshone the usual identification techniques by achieving both higher precision and lower false positives. The fuzzy testing module conveniently generates discriminative features while the CNN model provides trustworthy classification. This paper proposes a flexible and safe method to the disturbing problem of deception-aware adversaries detecting the honeypots in industrial CPS, but it has the assumption of pre-labeled data availability and the knowledge of error-handling signatures that can limit adaptability in dynamic or heterogeneous environments.

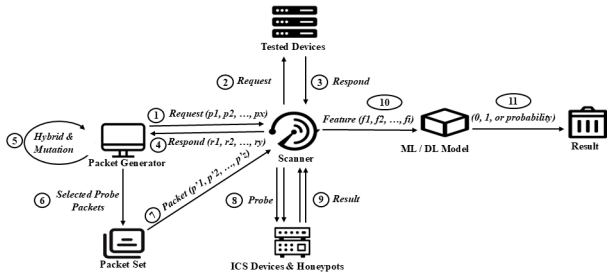

Fig. 9. The architecture of industrial CPS honeypot identification in [110].

In [111], the authors introduced a two-layer deception strategy for CPS. By depicting defender-attacker interactions as a signaling game with incomplete information, the model allows coordinated deception to be implemented both at the application layer (e.g., fake user interfaces) and at the network layer (e.g., falsified messages), which secures detection avoidance through consistency. The defender's strategies, communicating either honest or misleading signals, are improved using Perfect Bayesian Nash Equilibrium (PBNE), while the attacker checks the system type (normal/ honeypot) to determine whether to attack. Experiments carried out in an automotive CPS setting proved that two-layer deception not only increased the uncertainty of the attacker but also brought down the payoffs, and made the cost-deterrence more effective in comparison to single-layer techniques. This work, which is based on the Purdue Enterprise Reference Architecture, is the first to scientifically model multi-layer deception in CPS through game theory, thus presenting a solid framework against such attacks.

Table IV summarizes all the discussed works in cyber-physical systems, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

IX. Cyber Deception in Industrial Control Systems

Industrial Control Systems (ICS), which are specialized cyber-physical systems, are designed to monitor and control the operations of industries such as manufacturing, energy, water, and transportation. The components of ICS include SCADA systems, Distributed Control Systems (DCS), and Programmable Logic Controllers (PLCs). An ICS architecture typically consists of field-level devices, controllers (PLCs/RTUs), supervisory systems (SCADA), HMIs, and communication networks that use industrial protocols such as Modbus and DNP3. As ICS were originally designed for isolated, deterministic environments, they have gradually become interconnected with IT networks, which has led to the problems of legacy components, weak authentication, and insecure communication protocols. They are exposed to sophisticated cyberattacks

TABLE IV. SUMMARY OF CYBER DECEPTION TECHNIQUES IN CYBER-PHYSICAL SYSTEMS

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[107]	Control adversarial actions to align with system objectives by crafting information strategically	Advanced adversaries seeking system-related information	Strategic signaling (linear or "linear plus noise" rules) to manipulate adversary perception	Sender and receiver agents, Gaussian information, Quadratic cost functions	Covariance of posterior estimate, semi-definite programming (SDP) equivalence, stackelberg equilibrium outcomes	Limited to Gaussian information, non-classical information schemes in control settings, adversaries are rational
[108]	Optimize defensive strategies against network attacks in CPS by balancing honeypot deployment and human analysis costs	Network attacks (e.g., probing, exploitation of vulnerabilities)	Low- and high-interaction honeypots	Cyber layer (control systems, communication networks), physical layer (sensors, actuators, plants), honeypots (low/high-interaction), firewall, router, servers	Bayesian-Nash equilibria, defense payoff maximization, escape probabilities of attacks, human analysis cost allocation efficiency	Static attackers, static deployment, dynamic adaptation not addressed
[109]	Detect and block CPS-targeted cyberattacks using honeypots	DoS, spoofing, control hijacking, malware, code injection	Low- and high-interaction honeypot-based intrusion detection	Cyber layer (sensors, network), physical layer (actuators, processes), honeypot (virtual/physical), security rules engine, traffic monitoring system	Attack capture rate, scalability, interaction level	Single-system use, Static security rules may not adapt to evolving threats
[110]	Detect honeypots to protect deception infrastructure integrity	Honeypot fingerprinting in softwarized CPS	Fuzzy probe generation + CNN-based honeypot classification	Secure Fuzzy testing module, scanner, response analyzer, deep learning model, dynamic packet set	Accuracy/ precision/ recall, effectiveness, diversity of response packets	Labeled training data, focused on Modbus protocol
[111]	Strengthen CPS resilience by implementing a dual-layer deception strategy	CPS attacks across application/ network layers	Coordinated multi-layer deceptive signaling	Coordinated signal senders (application/ network), belief update model	Defender's payoff, deterrence effectiveness	Increased deployment complexity and cost, fixed attacker beliefs

like unauthorized command injection, manipulation of control logic, stealthy data tampering, and firmware exploitation. Stuxnet, Triton, and BlackEnergy incidents are examples of severe physical disruption and operational downtime through intentionally attacking ICS. Conventional security mechanisms generally do not have the contextual awareness, adaptability, or low-latency operation required in real-time ICS settings, and in addition, frequent patching as a remediation is often impractical due to availability and safety constraints [112]. In response, cyber deception is a good choice not only because it is a proactive and stealthy approach but also it is a strategic alternative such as inserting misleading elements, decoy sensors, fake field devices, deceptive control logic and protocol-emulating honeypots, in the ICS environment to mislead adversaries, trigger early alerts, delays attacker progress, and aid forensic investigations. Design of deception in ICS is a difficult case due to the necessity of preserving protocol fidelity, safety, and the need for realism under the threat of expert attackers. This section explores deception strategies specifically tailored to ICS, emphasizing their architectural integration, effectiveness against known threat models, and deployment limitations.

In [113], the authors suggested the implementation of Honeyd+, an applicable, scalable, and cost-effective implementation of high-interaction ICS honeypots through only one PLC and a proxy-based engine. For example, by changing the network/device identifiers (IP/MAC addresses, hostnames, serial numbers) dynamically and manipulating protocol payloads, Honeyd+ can simulate multiple physical PLC devices while preserving the culture of identity and realism. Tests on Raspberry Pi and high-end laptops certainly indicated its capacity to register more than 75 virtual networks, while, under other moderate traffic, it maintained low error rates, making it effective for reconnaissance deception. Though weaknesses include no support for formatted encrypted protocols, scalability constraints due to PLC features, and poor performance under extremely high traffic. The work shows possible improvements such as protocol-specific adjustments and the integration of diverse ICS components, to extend it to the production environment.

In [114], the authors presented HoneyPLC, a high-interaction honeypot that closely simulates programmable logic controllers (PLCs) in ICS. In contrast to conventional ICS honeypots such as being low interaction fidelity, limited protocol support, and poor malware capture, HoneyPLC, as illustrated in Fig. 10, is equipped with a modular framework that consists of a PLC profile repository, a personality engine (for protocol spoofing and fingerprinting), network services (HTTP, SNMP, S7comm), and an interaction data module to log attacks and capture malicious ladder logic injections. The key innovation is a PLC profiling toolkit that dynamically, for example, configures the Nmap fingerprints and SNMP MIBs to imitate real devices such as Siemens S7-1200 or Allen-Bradley MicroLogix 1100. Beyond that, HoneyPLC's S7comm server and TCP/IP stack operate realistically to trick tools such as Nmap, PLCScan, and Siemens Step7 Manager, whereas its ladder logic capture module records the injected malware for future analysis. The tests demonstrate that there is a high fidelity (>90% Nmap Confidence, Shodan scores matching real PLCs) and scalability (75+ simulated hosts per PLC). It is worth mentioning that the main shortcomings are the lack of physical process simulation and the fact that there is no support for encrypted protocols (e.g., Modbus Secure).

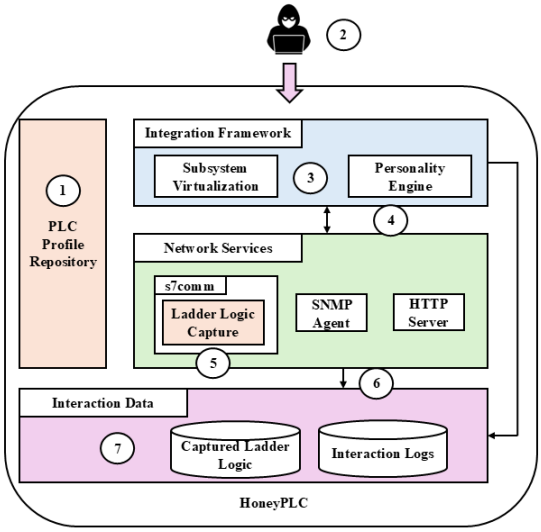

Fig. 10. The architecture of HoneyPLC in [114].

In [115], the authors introduced two deception methods for the identification of stealthy malware in the ICS, which avoid the traditional sensors through the exploitation of internal host discovery mechanisms such as NetBIOS browse lists and ARP tables. The methodology is based on the addition of sensor information to these lists in a way that causes malware to be attracted to self-disclosure. Proposal 1 is aimed at malware such as Conficker through the broadcasting of fake NetBIOS advertisements that are continuously emitted, which makes sure that the sensor's hostname is listed in the network browse lists, a tactic that is most advantageous under Windows-based systems, particularly in old versions of the program. In contrast, Proposal 2 incorporates the scheme which ensures that the sensor's IP-MAC pairing is embedded in the ARP caches of all hosts by sending cyclical ICMP or ARP requests, which, in turn, counteract malware that uses ARP tables for lateral movement. The techniques are designed for simple and non-intrusive application in the vulnerable ICS networks. The evaluations validate that Proposal 1 was successful in identifying Conficker in a laboratory setting, whereas the practicality of Proposal 2 is denoted in a theoretical proposal, which remains to be verified with real ARP-based malware. Despite their efficacy, the techniques, on the other hand, are based on assumptions relating to the behavior of the attacker, and involve a need for trial-scale deployment, decreasing the number of false positives, and integration with further detection frameworks.

In [116], the authors introduced a deception defense framework for ICS leveraging adaptive Hidden Markov Models (HMM) for the detection and mitigation of cyber threats, which include DDoS attacks, malicious program injection, and false data injection. The framework models typical ICS behavior using multidimensional data and detects anomalies by comparing observed behaviors with expected HMM state sequences. It adapts to evolving threats through incremental updates of HMM parameters. While primarily focused on anomaly detection, the framework, as depicted in Fig. 11, integrates a passive form of deception by analyzing attacker interactions within virtual environments, enabling misdirection without exposing the real system. Detected attacks (e.g., TCP SYN floods, ladder logic injections) trigger appropriate responses such as node isolation or administrator alerts. Through experiments conducted on an OpenPLC-based testbed, the solution was found to have very high rates of detection (98% for DDoS, 95% for malware injection, and 90% for false data attacks), very low percentages of false alarms (1-3%), and quick response times (1-2.7 seconds). Whereas the outcome of these tests was successful, it is mentioned that scalability and denial of multi-stage attack coverage are the shortcomings, which place emphasis on broadening the scope of the framework's deployment in larger platforms and its interfacing with other security orchestration platforms.

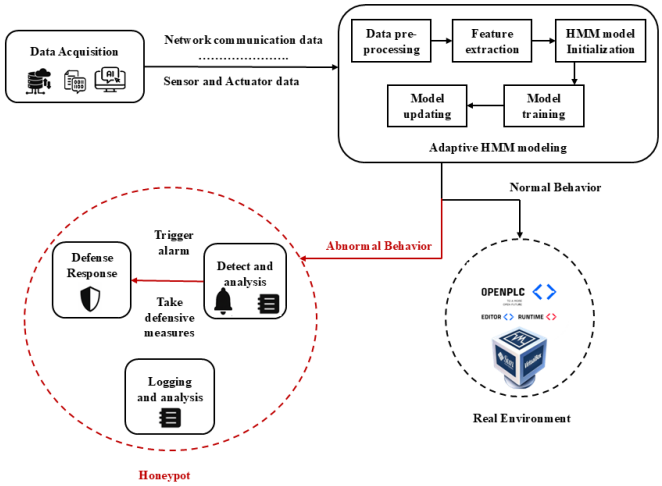

Fig. 11. The adaptive HMM-based ICS deception framework in [116].

Table V summarizes all the discussed works in industrial control systems, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

X. Cyber Deception in Smart Grids

Smart grids are the modernization of traditional electrical networks by integrating advanced information and communication technologies to make power systems more efficient, reliable, and sustainable. The systems comprise components like smart meters, sensors, automated control systems, and communication networks that together deliver the real-time monitoring and control of energy flow innovation. However, the high degree of networking and the dependence on digital technologies have made the trouble spots more and more numerous, and that is the reason the smart grids have become vulnerable to a strong variety of cyber threats including for example, false data injection attacks, denial-of-service attacks, and embedded malware attacks. The cyber-attacks on Ukraine's power grid, which were particularly prominent cases, showed the high impact of such vulnerabilities, bringing about the consequences of significant grid disruptions as a result of that so demonstrating the potential for major disturbances. The traditional measures of cybersecurity are often inefficient in the face of the ever-changing, elusive characteristics of these

TABLE V. SUMMARY OF CYBER DECEPTION TECHNIQUES IN INDUSTRIAL CONTROL SYSTEMS

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[113]	Create low-cost, authentic, and targetable honeypots for Industrial Control Systems (ICS) to detect and analyze attacker TTPs	Reconnaissance, enumeration, initial access, and persistent attacks (e.g., via TCP/UDP port scanning, protocol exploitation)	Proxy-based high-interaction honeypots with Honeyd+	Physical PLC, Honeyd+ host, search/ replace Engine, evaluation platforms	Cost-effectiveness, error rate, scalability	Limited support for encrypted protocols, high load handling, PLC bottlenecks
[114]	Understand attack methods targeting Programmable Logic Controllers (PLCs), by deceiving attackers into interacting with a honeypot	Malware injection (e.g., Stuxnet, Crashoverride), reconnaissance, and disruption of physical processes	High-interaction honeypot simulating real PLCs with advanced protocol simulations	PLC profile repository, personality engine, network services module, interaction data module, PLC profiler tool	Stealthiness, compatibility with proprietary tools, ladder logic capture	Limited protocol coverage, lack of physical interaction modeling
[115]	Detect malware infections in Industrial Control Systems (ICS)	Malware infections (e.g., WannaCry, Conficker)	Embeds sensor host names in browse lists, embeds sensor IP addresses in ARP tables	Sensor, host announcement module, ARP/ICMP module, alert system	Detection capability, false positives, coverage vs. traffic	Deployment impact (Proposal 2's continuous ARP requests may affect network performance), evasion by advanced malware, protocol support
[116]	Protect ICS from cyber-attacks by detecting and responding to anomalous behavior in real-time	DDoS attacks, malicious program injection, false data injection	Adaptive Hidden Markov Model (HMM) for modeling normal behavior and detecting anomalies	Data acquisition, adaptive HMM, honeypot-based response, OpenPLC testbed	Detection rate, false positive rate, response time	Dependence on HMM training data, multi-stage attacker handling

threats [117]. For that reason, a new type of defense was introduced, which involves cyber deception. By using deceptive elements like honeypots, decoy systems, and wrong data, cyber deception aims at disorienting attackers, slowing them fast, and earlier recognizing malicious activities. The implementation of cyber deception technology in smart grids, however, gives rise to difficulties, such as the assurance of component deceptiveness, maintenance of system performance, and seamless integration of deception strategies into the already existing infrastructures. In this section, the subject of cyber deception in smart grids is studied by considering methods, advantages, and limitations that the implementation brings.

In [118], the authors discussed the application of honeypot-based defense scheme to deal with the Distributed Denial of Service (DDoS) attacks aimed at the Advanced Metering Infrastructure (AMI) in the smart grids using a game-theoretic approach. The model suggested by the authors envisions the AMI as a tree like network consisting of smart meters, aggregators, and a central headend system. It is shown in Fig. 12 that the honeypots are to be embedded inside the firewalls of the WAN segment to make the traffic divert and to secure the headend and the back office servers. This strategy is framed formally as a Bayesian honeypot game, considering the presence of the users, the attackers, and the service providers, which include the strategies of the attackers, the effectiveness of the honeypots, and the anti-honeypot mechanisms that can be utilized. The model gives the specification of the payoff function and the resolution of the problem, finding multiple Bayesian Nash Equilibria (BNEs), which typically imply the optimization of the placement of honeypots so as to reduce energy consumption and improve detection rates. In the process of verification by experimental setups employing an OPNET-based AMI testbed, it has been established that the model surpasses the others in terms of higher detection rates (up to 85%) and more stable energy efficiency. The model operation deals with what is typically assumed as a rational and static attacker and simply presumes a deployment scenario.

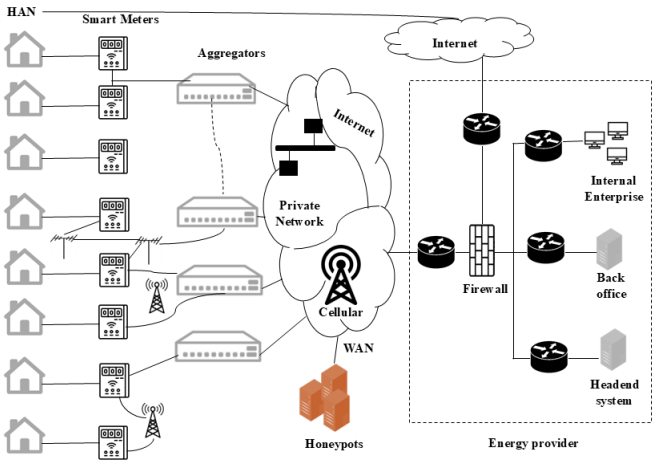

Fig. 12. The AMI network infrastructure with honeypot deployment in [118].

In [119], the authors thoroughly discussed and analyzed the honeypot and honeynet technologies that are utilized in the smart grid for security purposes. They have done an analysis of the factors such as the interaction levels (low versus high), purposes (production versus research), and deployment models (virtual versus physical) while evaluating the protocol supported by the industrial standards such as Modbus, IEC-104, and IEC-61850. The investigation has listed Conpot as the most versatile because of its modular design and multi-protocol capabilities as compared to twelve honey-x solutions. Even though Conpot is lacking in native support for advanced protocols like GOOSE/MMS, DNP3, it is the most versatile

choice. The EU SPEAR project has set up a smart home testbed, which implemented a practical example of Conpot. Conpot has been proven to be effective by applying modified iptables rules and JSON-based logging in the process, while also outlining its limitations in high-interaction fidelity, IPv6 adoption, and dynamic threat adaptation.

A two-phase investigation into enhancing threat intelligence collection for smart grid systems using honeypots was conducted. In the first study [120], low-interaction honeypots were implemented across five AWS regions (Singapore, US, Canada, Germany, Brazil) to simulate ICS protocols IEC 61850, IEC 60870-5-104, DNP3, Modbus TCP, and BACnet. They succeeded in remaining undetected by fingerprinting tools like Shodan's Honeyscore while collecting a total of 6GB of network traffic over six months. Their analysis shows indicators of probing behavior specific to the protocol, geographical correlations on scanning (with 0.97 cross-correlation between Germany and Brazil instances), and the presence of persistent threat actors such as 54 IPs that were detected monthly and executed SYN-flood attacks in a coordinated manner, all of which provided a valuable reference for IDS and firewall tuning. On the other hand, the study has some limitations, such as the fact that the honeypots are of low interaction, which leads to a lack of realistic behavior, and that post-compromise attacker activities are unable to be captured, which is a strong reason to think about the use of more complex, high-fidelity honeypots in other smart grid security studies. To address these limitations, their second study [121] demonstrated a realistic smart grid honeypot system by creating a full-scale infrastructure that includes SCADA HMIs, historians, IEDs, protocol gateways, and firewalls. It employs a decoupled architecture that is integrated with various tools such as Honeyd for anti-fingerprinting, Cowrie for SSH deception, SoftGrid for power-flow simulation, and Mininet for virtual IEDs. The system achieves practical protocol spoofing (e.g., IEC 61850 MMS) and cyber-physical consistency through these tools. Security testing has proven the system's capability to fingerprint devices strongly, log multi-phase attacks by using transparent proxies, and be resistant to probing by adversaries. Yet, difficulties persist in lateral movement modeling, automating dynamic deception, as well as adapting to the strategies of the attacker.

In [122], the authors proposed DecIED, a scalable deception framework for smart grid substations that is based on the idea of k-anonymity to conceal the identity of actual Intelligent Electronic Devices (IEDs) by introducing several indistinguishable decoy IEDs. Tailored for the IEC 61850-compatible environments, DecIED mimics the actions and communication patterns of the genuine devices by implementing the protocols, such as MMS, GOOSE, and SV. The outline, as shown in Fig. 13, involves a virtual IED program which consists of an IED server and logic, GOOSE/SV subscribers and publishers that are all integrated with Honeyd to spoof OS fingerprints and port behaviors. Each decoy IED is provided a unique IP/MAC address, and by following the same timing and content of the messages, it can replicate real devices. The system is designed to be resource-efficient, being able to commit over 200 deception instances on a single industrial PC; thus, scalability is achieved without compromising fidelity. Evaluations prove that deception was achieved without the capacity of real devices in passive and active reconnaissance. Indeed, DecIED out of the box maintained realistic control logic responses and nevertheless succeeded in the deception of scanning tools. Though issues prevail in the process of really exacting OS-level functionality and diverse logic types problems not addressed, yet DecIED is a solid ground for deception in today's substations and, it can be a stepping stone for further tactics like moving target defense.

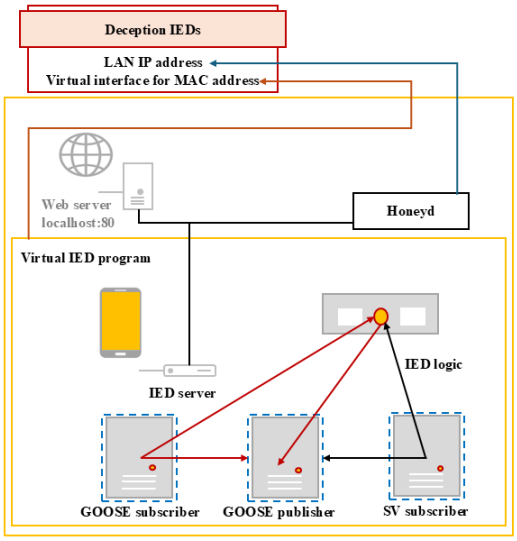

Fig. 13. The module architecture of DecIED in [122].

In [123], the authors proposed a defense strategy framework for smart grids with the help of the honeypot model. This is made with an incomplete information stochastic game that is used for describing the dynamic interactions between the attacker and the defender. The system consists of low-interaction and high inside honeypots in the networks like smart grid servers (web, FTP, OPC, and application servers), where it is isolated through firewall rules. A six-tuple game model that describes the defense mechanism is set up partly by attack/ defense actions, privilege escalation states, transition probabilities, and host-specific metrics (e.g., importance and risk level). The model of Nash equilibrium is obtained by solving it through Gambit software, and as a result, the framework comes up with state-optimal defense strategies that enable adaptation to the attacker through privileges or honeypots. The evaluations show that they achieved good deception rates and resource allocation, yet the model was derived assuming the topology is static and the state transitions are known perfectly, which in turn limits its scalability in a dynamic environment.

Table VI summarizes all the discussed works in smart grids, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

XI. Cyber Deception in Internet of Things

The Internet of Things (IoT) is one of the numerous interconnections that link up physical objects, devices such as sensors, actuators, gateways, and cloud-based analytics platforms, which will work together to monitor, control, and automate different domains like healthcare, smart homes, industrial systems, and smart cities in real time. Rendering a typical IoT

TABLE VI. SUMMARY OF CYBER DECEPTION TECHNIQUES IN SMART GRIDS

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[118]	DDoS mitigation in AMI networks	Distributed Denial of Service (DDoS)	Honeypot-based game-theoretic decoy placement	Smart meters, data aggregators, central head-end, honeypots behind firewalls	Detection rate, energy efficiency, BNE solution space	Rational attackers, limited to small-scale testbeds
[119]	Secure smart grids by diverting attackers, collecting threat intelligence, and preserving forensic evidence	Broad coverage, including SCADA exploits, DDoS, and malware targeting ICS protocols (Modbus, IEC-104)	honeypots, honeynets	Emulated ICS devices (PLCs, RTUs), protocol servers (Modbus, SNMP), virtual/ physical honeypot nodes, tools (Conpot, GridLAB-D (power flow simulation), Mininet (SDN))	Detection Rate, realism scalability	Protocol gaps, resource intensity, dynamic threats
[120]	Collect threat intelligence on smart grid attackers	Scanning/probing, SYN-flood DoS, protocol specific exploits (IEC 60870-5-104, DNP3, Modbus)	Low-interaction honeypots emulating ICS protocols	AWS instances, TCP listeners, wireshark, ELK stack	Packet counts (TCP/ICMP), geographic correlation of attacks, source IP persistence	Lacks high-fidelity engagement, low-interaction design limits realism of emulated devices, limited protocol coverage
[121]	Collect threat intelligence, delay attacks, and deceive attackers into engaging with the honeypot	Attacks via VPN/SSH/RDP, malicious IEC 60870-5-104/IEC 61850 commands, lateral movement	High-fidelity honeypots with OS fingerprint spoofing and decoupled logging	Control center (SCADA HMI, Historian DB, VPN Server), substation gateway, Virtual IEDs, external (Jumpbox, transparent proxies)	Realism (Nmap/P0f fingerprint similarity to real devices, detection of virtualization), logging (captured attack phases)	Passive devices: IEDs (passive servers) easier to mimic than active devices (PLCs/ gateways)
[122]	Detect reconnaissance activities, prevent attackers from identifying real IEDs	Passive monitoring, active probing, lateral movement	*k*-anonymous smokescreen (decoy IEDs)	Honeyd (anti-fingerprinting), virtual IED module, Nginx web server, transparent proxies	Indistinguishability ratio, device similarity, scalability	Imperfect OS fingerprinting, requires access to real IED logic for full imitation
[123]	Optimize defense strategies against smart grid attacks	Multi-stage attacks with privilege escalation	Honeypot deployment guided by stochastic game model	Web/ FTP/ OPC/ application servers, honeypots, firewall rules, attack/ defense action sets	Game-theoretic Nash equilibrium, host significance, security risk level, state transition probabilities, fefense success rate	Convergence time, requires predefined vulnerability data

architecture model, one can think of it as a layered framework, for example, comprising the perception layer (sensors and actuators), the network layer (Wi-Fi, Zigbee, 5G), and the application layer (cloud or edge services). Be that as it may, IoT systems are attacked with major security problems such as device heterogeneity, resource limitations, root defaults, and lack of standardization. Such weaknesses permit IoT networks to come under attack from an array of cyber threats, such as device hijacking, eavesdropping, man-in-the-middle attacks, and DDoS amplification through botnet devices like Mirai. In such constrained and distributed environments, traditional defense mechanisms, like encryption and authentication, often seem to be inefficient [124]. Therefore, the cyber-deception approach has been chosen as the effective way. Now defenders deploy honeypots, fake sensors, deceptive APIs, or synthetic data flows to mislead the adversaries, gather threat intelligence, and delay attacks. Among the recent frameworks that have emerged are AIIPot and IoTFlowGenerator that exploit machine learning to synthesize realistic decoy behavior and traffic patterns; thus, they increase IoT deception fidelity. However, IoT is full of particular challenges in the deployment of effective solutions for example the need for major realism at scale, load overhead maintenance on the constrained devices, and the need for seamless integration of deception into the edge/cloud management platform. This section is focused on the deception techniques that are designed specifically for IoT environments, mainly dealing with component-level placement and adaptive threat engagement.

In [125], the authors introduced HoneyCloud, which is a scalable honeypot framework that is meant for analyzing fileless attacks on Linux-based IoT devices. The setup employs four hardware honeypots (like Raspberry Pi, BeagleBone) and 108 software honeypots distributed over eight public clouds. Compared to other hardware honeypots, they are superior in terms of high-fidelity interactions. However, the cost and maintenance overhead issues drove the authors to use QEMU-based software honeypots running OpenWrt, which were enhanced with realism-preserving techniques (like CPU masking, bus emulation). Over 12 months, HoneyCloud amassed 26 million attacks, including 1.5 million fileless attacks, which made it possible to create the first taxonomy of IoT fileless threats such as credential theft, configuration tampering, and SSH tunneling. The major findings illustrate that 65.7% of fileless attacks utilize default shell commands (like rm, kill, passwd) and demonstrate a trade-off: read-only filesystems are resistant to malware persistence, but they also inhibit shell-history auditing for fileless detection. The paper advances IoTCheck, a workflow for defense that entails the hardening of devices, though shortcomings are the inability to decrypt SSH tunnels and absence of support for newer IoT interfaces (like ZigBee). HoneyCloud broadens the perception of stealthy, non-malware threats, as these are the kinds of threats that can evade traditional defenses.

In [126], the authors analyzed empirically the IoT botnet

behavior by using three medium-interaction Cowrie honeypots programmed to emulate SSH/Telnet services and IoT file systems. The honeypots (which were scaled with Apache/Postfix services and ELK stack logging) observed for more than 40 days and recorded malware downloads during login sessions, which were later clustered by the Levenshtein edit distance on command sequences. According to this study, the majority of sessions that were related to malware were from Mirai variants at 97% with the loader patterns being two dominant ones (74.3% and 23.4% prevalence). The authors can confirm that these specific IP addresses operate within a small group of loaders distributing various strains of Mirai, which are often just slightly modified (e.g., by using UPX packing, filler commands, or compiler flags). Furthermore, the system registered some threats that were scarce (e.g., IRC botnets), although the limitations included Cowrie's static emulation (which was lacking high-interaction behaviors), being biased towards x86/64 architecture, and not being able to analyze non-file-downloading sessions.

In [127], the authors presented a scalable and hybrid architecture for IoT honeynet, which consists of three types of honeypots to enhance attack capture and deception fidelity: a medium-high interaction honeypot emulating the CVE-2017-17215 vulnerability, a high-interaction honeypot using real IoT firmware to handle unprocessable requests, and a multiport honeypot simulating SOAP services across exposed ports (e.g., 37215, 52869) with dynamic banner generation based on real device fingerprints. As shown in Fig. 14, the system is distributed over several physical nodes, each one being managed by a local master program and coordinated through a central control node. The architecture takes advantage of Docker both for fast deployment and isolation, being complemented with self-check mechanisms that ensure honeypot stability. Traffic is first routed to medium-interaction honeypots and deployed to high-interaction nodes if a deeper emulation is required. Deployed in the US and Canada, the honeynet remained stable for six months, capturing 332 unique attacker IPs and malware samples that went undetected by VirusTotal (e.g., "Ixa", "gyv"). Geographical analysis disclosed that scanning behaviors stemmed mainly from Japan and injections from the US. Although they are effective, the authors still mark the limitations in the automation issue and post-compromise behavior modeling and propose future work in these areas.

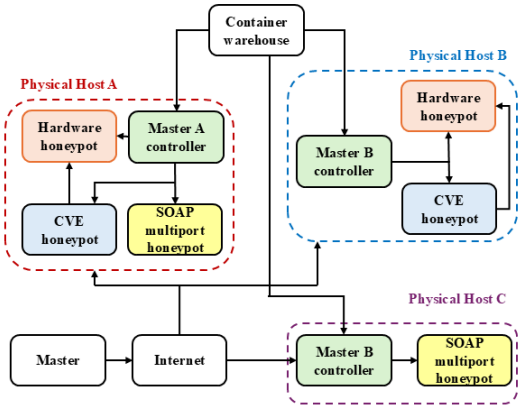

Fig. 14. The composite honeynet system architecture in [127].

In [69], the authors introduced a proactive defense framework, which is a software-defined networking (SDN) enabling technology specifically developed to combat Distributed Denial-of-Service (DDoS) attacks in IoT networks by using the integration of Moving Target Defense (MTD) and cyber deception techniques within a programmable architecture. The framework, as illustrated in Fig. 15, consists of three main parts: the first one is the monitoring agent who is responsible for the detection of attack behaviors, and the second one is the decision module which utilizes a multistage signaling game for modeling the attacker-defender interactions and predicting the adversarial strategies, and the last one is the deployment module that carries out such deception tactics such as random traffic delays, rerouting, and IP randomization. The system can deceive attackers through the use of a switching mechanism that dynamically changes the network properties and thereby alters the perception of the attacker. The manipulation includes operating under true services or honeypots depending on the behavior and confidence of the attacker. The signaling game analysis generates Perfect Bayesian Nash Equilibria (PBNE), with the Optimal Strategy Algorithm (OSA) on top of that, which is used to optimize defense actions that are under the condition of uncertainty. Performance evaluations on both Mininet and a real SDN testbed also show that the improved method, besides survival rates, has controlled CPU usage in the range of 65%, and reduced latency and packet loss when compared to baseline measures. Although it is a very efficient framework, the improvement should be the deployment of security mechanisms and assets that are adaptable and proactive to combat insider threats, which are usually tough to differentiate from legitimate users.

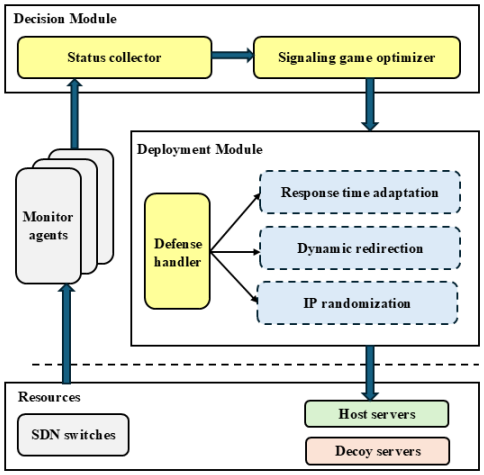

Fig. 15. The proposed framework in [69].

In [128], the authors introduced a cyber deception framework especially designed to fight epidemic botnet propagation in IoT networks using game-theoretic approach. They represented the interaction between attackers and defenders as a zero-sum, one-sided Partially Observable Stochastic Game (OS-POSG). In the proposed model, the defender places some honeypots strategically to monitor and disrupt botnet communication, and the attackers, in turn, know everything about the network structure except the actions of the defender, and their primary goal is to maximize the number of infections. The

defender's partial observability comes from being notified of the reactions of IoT users only (e.g., patching or password changes). Two deception strategies are examined, which are: a randomized deception strategy (RDS) and a k-smart deception strategy (k-SDS), where honeypots are positioned based on the infected devices. The attacker uses straightforward strategies (unicast, half, or broadcast) instead of Q-learning. Simulations on synthetic IoT topologies measure the maximum proportion of infected devices and Time to Extinction (TTE). The results show that k-SDS is significantly better than RDS, especially in high degree connectivity networks. The framework is basically an affirmation of the power of the dynamic deception that is applicable under partial observability. Further actions include the derivation of Nash Equilibrium strategies and the exploration of state-dependent deception techniques.

In [129], the authors presented HoneyComb, a cyber deception framework that has a proactive character and combines a /8 darknet network (16.7 million IPs) with mitigating IoT malware forensic artifacts at scalability. The main distinguishing feature of HoneyComb from the traditional honeypots, which passively wait for attacks, is that it actively detects malware-infected IoT devices by inspecting unsolicited darknet scans and sends then crafted TCP SYN-ACK packets to exploit stateless scanning flaws in IoT malware (e.g., Mirai's dst.IP == TCP.seq signature). The structure, as illustrated in Fig. 16, contains: 1) a darknet traffic analyzer (CAIDA data), 2) a cloud-based honeypot interface, and 3) a pool of virtualized ARM-based IoT devices (OpenWrt) emulating Telnet/SSH services. Once the connected infected devices are tricked, HoneyComb logs malware binaries (1,398 unique URLs), HexString dump commands, C&C communications, busybox commands, and credentials. Throughout a 48-hour evaluation period, HoneyComb reported a total of 1.4 million interactions, which were successfully carried out by 37,323 infected devices, attributing them to 40+ malware variants (e.g., Hajime, Mozi, ZHTRAP). Noteworthy statistics show a 22.07% response rate (37K/169K scanned IPs) and 90% of responses in 750ms or less. Some of the limitations are: 1) ARM-only virtualization (excluding MIPS/x86), 2) low SSH engagement (5.55% vs. Telnet's 25.88%), and 3) proxy/NAT evasion, and improvements will support a broader range of CPUs, simulate other services, and improve proxy penetration, ultimately achieving a real-time IoT malware knowledge graph for collaborative defense.

In [130], the authors introduced IoTFlowGenerator, which is a cyber deception framework based on deep learning that synthesizes realistic IoT traffic flows to increase honeypot believability against adversaries who monitor network activity. Being different from traditional IoT honeypots, which lack the real traffic patterns and can be easily detected, IoTFlowGenerator employs a fine-tuned SeqGAN and VQ-STAE (Vector Quantized-Sequence Transformer AutoEncoder) to produce multivariate, time-consistent packet metadata that imitates real user-device interactions. The system processes PCAP files (using Wireshark) to extract features such as packet length, direction, and timing, clusters packet-level signatures using DBScan, and encodes packets into discrete tokens. These tokens are adversarially trained into synthetic sequences, which are then reconstructed into traffic flows with noise injection for realism. Using 18 IoT devices for evaluation, IoTFlowGenerator not only outperforms DoppelGANger but also reduces the success rate of the adversarial ML models in the differentiation of synthetic traffic from real traffic. Particularly, this reduction is in synthetic-data-aware attack scenarios. Despite its high potential, the main weakness is the dependence on pre-collected traffic data and device-specific modeling, which might pose scalability challenges. The research highlights how sequence-based GANs can be used to increase the deception of encrypted IoT environments.

Table VII summarizes all the discussed works in internet of things, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

XII. Cyber Deception in Internet of Vehicles

The Internet of Vehicles (IoV) is a transforming new phase of intelligent transportation systems, which is now become possible due to the distinct features of the network that allows vehicles to communicate with each other (V2V), with infrastructure (V2I), and with broader networks (V2X) thus improving safety, efficiency and user experience. During this time, increased connectivity has also brought security problems, which are challenging to solve. Protection measures against the spectrum of attacks facing IoV systems, such as spoofing, eavesdropping, denial-of-service attacks, and unauthorized access, have compromised both data integrity and physical safety. Traditional security measures are not capable of tackling the challenges of these compound threats [131]. Furthermore, recent works [132] [133] reported the provision of intelligent sensors in IoV via reinforcement learning based techniques, where virtualized sensors are dynamically assigned to achieve the best possible service and resource optimization. Nevertheless, the main issue of security in IoV persists, and such provisioning schemes must be examined about their potential role in supporting defensive strategies such as cyber deception, which has emerged as a proactive defense strategy. Through the use of such deceptive elements as honeypots, decoy nodes, and faulty communication protocols, the defenders can mislead attackers, make them late, and procure important information on their strategies. We have the latest findings that bring sophisticated frameworks to the IoV setting. To illustrate, the DECEPTWIN framework leverages digital twins and blockchain to create a decoy environment that looks real, enhancing the detection and analysis of malicious activities. Similarly, Honey-Car framework is a great example of the use of game-theoretic models in the optimization of honeypot settings; thus, the attackers are involved well, and we can take actionable threat intelligence. Even if these ideas are great, the process of introducing cyber deception into IoV has its problems. It is vital to create decoys that are real, as well as sustain system performance and incorporate strategies of deceit in a smooth manner into the already established setup. This section considers the cyber deception play in IoV, looking into its methodologies, advantages, and the issues raised during the process of execution.

In [134], the authors introduced NHBADI, a newly introduced honeypot-based scheme for identifying and isolating Black Hole attacks in MANETs. The system utilizes AODV protocol modification so that it sends out RREQ packets to nonexistent nodes with the help of the destination IDs and TTL=1 Flag; thus, the system identifies the malicious nodes

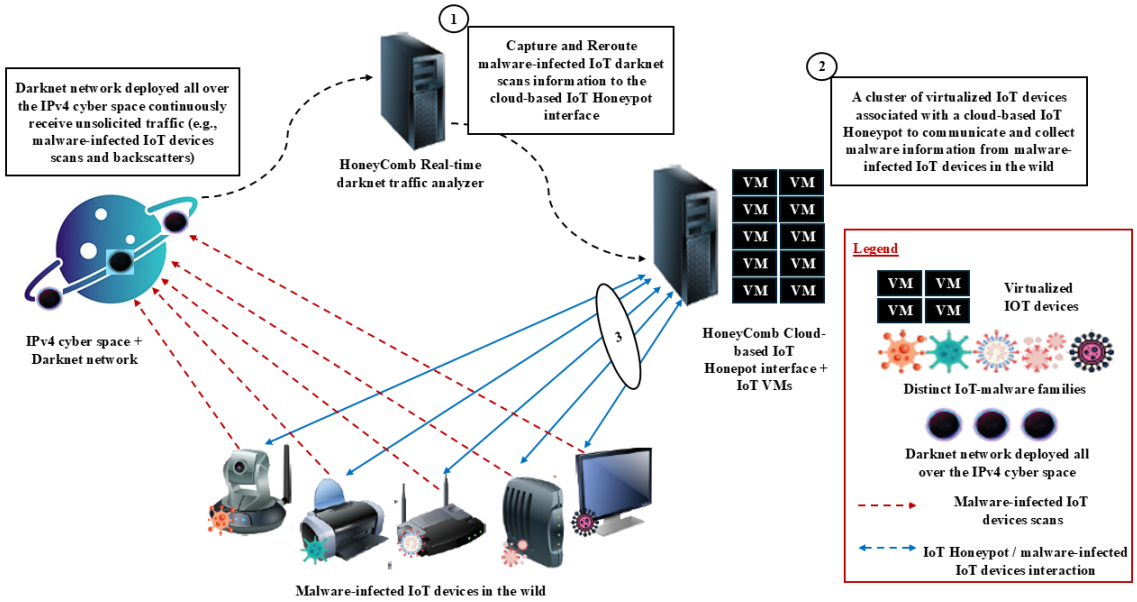

Fig. 16. The overall architecture of the proposed HoneyComb in [129].

TABLE VII. SUMMARY OF CYBER DECEPTION TECHNIQUES IN INTERNET OF THINGS

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[125]	Understand and defend against fileless attacks on Linux-based IoT devices	Fileless attacks (e.g., SSH tunneling, system data damage, de-immunization)	Hardware and software honeypots (HoneyCloud) with high fidelity emulation	Hardware/ software honeypots, shell interceptor & inference terminal, access controller, reset manager	Number of suspicious connections, number of effective attacks, cost comparison	No ZigBee support, visibility limits for encrypted tunnels
[126]	Track and analyze variations in IoT botnets (e.g., Mirai variants)	Mirai-based botnets (SSH/Telnet), SSH brute-force attacks, IRC-based botnets, coin-mining malware	Cowrie-based medium interaction honeypots	Cowrie honeypot, apache webserver, postfix mailserver, ELK stack, filebeat	Edit distance clustering, IP/file hash mappings, ssdeep similarity, session success rate	Limited to x86/64 binaries (Cowrie's emulated architecture), ssdeep inaccuracies
[127]	Capture and analyze IoT-specific attacks (e.g., exploiting UPnP/ SOAP vulnerabilities)	SOAP/UPnP-based attacks, port scanning, CVE-2017-17215 exploitation	Medium-high interaction honeypot, high-interaction honeypot, multiport honeypot	Honeypot core, docker containers, ELK stack, control center, QEMU	Attack volume, geographic distribution, malware samples, functional validation	Limited protocol coverage, static emulation, dependency on Docker
[69]	Proactively mitigate DDoS attacks in IoT networks by dynamically optimizing defense strategies	DDoS, attacks launched via IoT botnets (e.g., Mirai)	MTD techniques (IP/port shuffling, dynamic redirection, response time adaptation), honeypot	Monitor agents, decision module, deployment module, SDN controller	Survival rate, time Overhead, performance metrics (throughput, packet loss rate, CPU utilization, Round-Trip Time, requests per second)	Limited for powerful attackers, limited insider threat handling
[128]	Mitigate botnet propagation in IoT networks	Epidemic botnet (e.g., Mirai) launching DDoS	Honeypot deployment with strategic placement (e.g., k-SDS, RDS)	Defender (honeypot allocator), attacker (botnet propagator), IoT devices (S/I/R states), network topology (low/high connectivity)	Botnet Time to Extinction (TTE), maximum proportion of infected devices, defender's utility (recovery vs. infection rate)	Limited to random/ simple attacker strategies, assumes static game strategies
[129]	Proactive collection of IoT malware forensic artifacts to analyze and mitigate threats	IoT botnet propagation (e.g., Mirai variants)	Darknet-powered SYN-ACK deception with backend honeypots	Darknet traffic analyzer, virtualized IoT devices, cloud-based honeypot interface	Response rate, artifacts collected, response time, geographical/ASN distribution of infections	Limited to ARM architecture, limited protocols, proxy evasion
[130]	Generate realistic synthetic IoT traffic flows to deceive attackers into mistaking honeypots for real devices	Passive network sniffing to identify IoT device types	GAN-based synthetic flow generation (SeqGAN, VQ-STAE)	Traffic analyzer, VQ-STAE autoencoder, SeqGAN, packet fuzzing	Adversarial classification accuracy, traffic fidelity, response variability	Device-specific modeling, dependence on packet-level signatures

when they reply to these decoys. As for the results of the simulation in NS-2, NHBADI provides a good performance with 89.03% of the packets being delivered after the intervention (67.73% of packets with vulnerable AODV) and a routing load which is normalized to just 0.62 (110.79 in AODV with 4 Black Holes), in addition to these, the protocol has cut the end-to-end delay by 23%. The three-layer architecture (detection, verification, and isolation) efficiently quarantines attackers but is built on the assumption that they will react to the specific requests. The strategy might thus be limited in its effectiveness to more advanced adversaries. Through this, the authors show that tactical deception can actually be utilized to improve the security of MANETs more effectively with less effort in spite of the initial resource cost.

In [135], the authors provided a detailed survey on Intrusion Detection Systems (IDS) for Vehicular Ad Hoc Networks (VANETs) and VANET Cloud, dealing with such issues as dynamic en-route node resources, high mobility, protocol differences, and security threats (e.g., Denial of Service (DoS)). The paper condenses into various categories requesting IDS by deployment (distributed, centralized, hybrid), detection techniques (signature-based, anomaly-based, watchdog), and validation strategies while reviewing a total of nearly 25+ different alternatives in a critical manner to achieve trade-offs on accuracy, latency, and overhead. The proactive Honeypot-Optimized IDS (HPIDS) is prominently and solely presented here, which uniquely incorporates bait-based deception along with lightweight detection in order to defend both known and zero-day threats. The article enlists some challenges left unresolved, including a high number of false positives, mobility resilience, and VANET Cloud-specific gaps. Meanwhile, it suggests the implementation of adaptive, low-latency frameworks, along with the privacy of personal data preserved.

In [136], the authors introduced HoneyCar, a game-theoretic framework dedicated to the optimal honeypot configurations in the IoV context, by modeling adversarial interactions as a repeated imperfect-information zero-sum game between a Defender (network administrator) and an Attacker. Using vulnerabilities data from CVE database and CVSS metrics, HoneyCar in its evaluation explores two scenarios: HCG-a, which ignores reconfiguration costs and considers maximal engagement of the attacker through Low-Interaction Honeypots (LIH), and HCG-b, which incorporates reconfiguration costs to optimize cyber threat intelligence using High-Interaction Honeypots (HIH). According to the results of a case study that belongs to LIH, it is displayed that this option can significantly cost-effectively make an attacker waste time, while HIH, along with the selective reconfiguration, can reach the maximum intelligence gain. The framework is designed to be in equilibrium regarding deception costs and security benefits, while improvements are directed towards adaptive strategies and real-time reconfiguration.

In [137], the authors designed DECEPTWIN, a proactive deception framework for the IoV, which is a combination of Digital Twins (DTs) and blockchain, to create realistic, interactive decoys for attackers while securely logging their tactics. The architecture of the framework, as illustrated in Fig. 17, consists of six layers: 1) internet/networking for baiting via decoys, 2) physical system modeling with High-Fidelity DTs, 3) deception environments using Low-Fidelity DTs (LDTs) embedded with deceptive elements to mimic compromised IoV networks, 4) monitoring/tracking of attacker actions across kill-chain stages (reconnaissance, weaponization, exploitation, command and control, exfiltration, and covering tracks), 5) analysis/reporting of threat intelligence, and 6) blockchain-backed storage for immutable logs. Fig. 18 illustrates that DECEPTWIN, after understanding the attacker's interaction, adapts LDTs that show attacker interactions, all through deceptive commands, decoy communications, and falsified logs, enhancing realism while gathering actionable threat data. While it promises proactive IoV security, the framework remains conceptual; its scalability and practical implementation limitations are emphasized as major challenges for future work.

Table VIII summarizes all the discussed works in internet of vehicles, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified limitations.

XIII. Cyber Deception in Unmanned Aerial Vehicles

Unmanned Aerial Vehicles (UAVs) are widely used in various fields such as observation, transportation, and military. They include various devices inside like sensors, flight controllers, communication modules, and ground control stations (GCS). Hence, these can be the targets for hackers and cyber attacks, causing problems like GPS spoofing, jamming, or unauthorized access that result in compromised mission security and safety issues. Conventional protective measures are often insufficient to tackle these dynamic threats [138]. As a result, cyber deception has been developed as an effective solution. Counter-measures have been put in place such that by joining the honeypots with decoy nodes, the defenders can mislead the attackers, postpone their success, and collect useful data on their techniques. Recent developments have brought forth the use of clever deception schemes designed specifically for UAV environments. For example, the HoneyDrone framework is based on game-theoretic models to optimize the honeypot setups and to engage the attackers effectively with the goal of collecting actionable threat intelligence. Even with all the potential, integrating cyber deception in UAV systems is a tough process involving the identification of realist-looking decoys, performance continuity, and the difficulty of intertwining the deception tactics into pre-existing frameworks. This section discusses the utilization of cyber deception within UAVs, outlining its strategies, advantages, and the limitations experienced in its use.

In [139], the authors presented HoneyDrone, the first honeypot created specifically to simulate Unmanned Aerial Vehicles (UAVs) and effectively detect attacks targeting drone protocols. This medium-interaction honeypot also supports UAV-specific protocols such as Telnet, SSH, FTP, and the essential MAVLink standard, in addition to its low-cost deployment option on devices like Raspberry Pis. Its architecture comprises the following key features: 1) a Network Interface Emulator (NIE) which allows the simulation of wireless interfaces (Wi-Fi/Bluetooth/SiK); 2) a UAV Emulation module like a real drone with visualization of protocol behavior, filesystems, and telemetry; 3) MongoDB logging backend; and 4) Ardupilot SITL simulator for the realistic MAVLink flight data emulation. Two attack scenarios, the Telnet-based

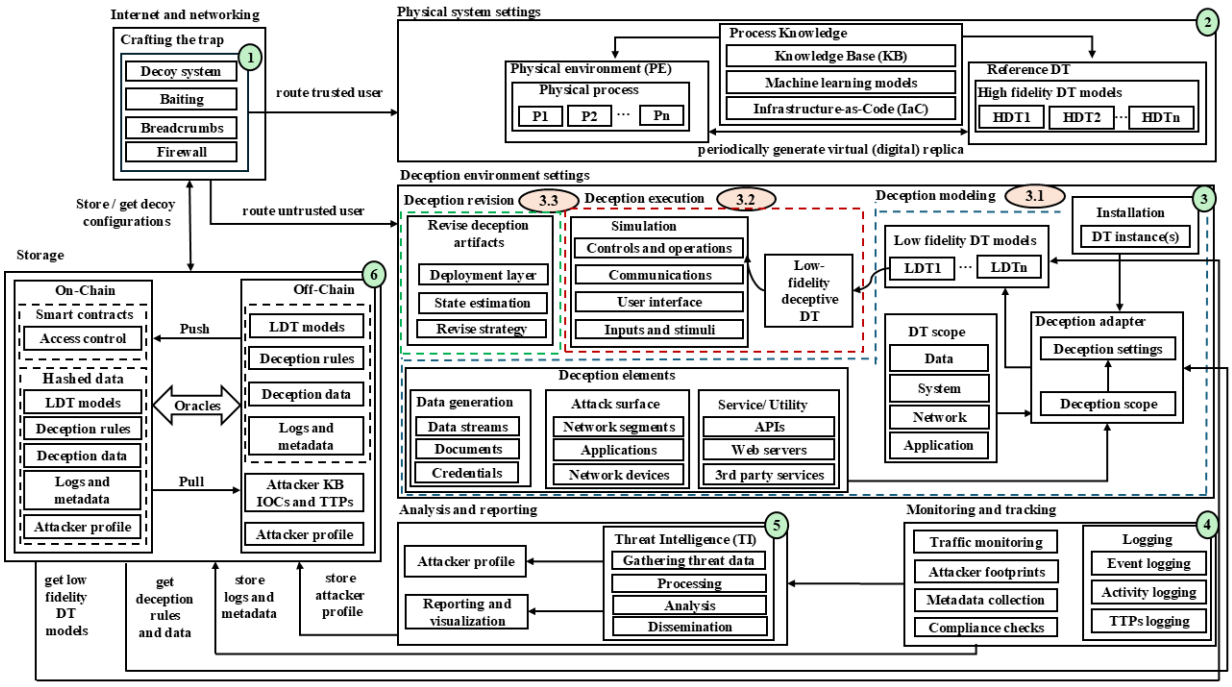

Fig. 17. The proposed architecture of DECEPTWIN in [137].

TABLE VIII. SUMMARY OF CYBER DECEPTION TECHNIQUES IN INTERNET OF VEHICLES

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[134]	Detect and isolate Black Hole attacks in MANET/IoV	Black Hole attacks (forged RREP messages in AODV)	Honeypot-based spoofed RREQ packets	Malicious node detection layer, route lookup layer, isolation layer	Packet Delivery Fraction (PDF), Normalized Routing Load (NRL), End-to-End Delay (EED), Packet Drop Ratio (PDR)	Assumes passive attackers, limited to AODV protocol
[135]	Detect and mitigate malicious nodes and attacks in VANETs and VANET Cloud	DoS, DDoS, Hidden Vehicle, Tunnel, Wormhole, Blackhole, Location Spoofing, Privilege Escalation, alteration, sybil, social Attacks	Honeypot-optimized IDS (HPIDS)	Signature-based IDS, Honeypot nodes, rule generation engine, distributed detection	Detection rate, false positive rate, detection time, resource overhead	Limited VANET Cloud-specific solutions, challenges in optimal placement and density of honeypot nodes
[136]	Develop deception strategies in IoV to engage attackers, gather intelligence, and optimize honeypot use within budget limits	CVE-based exploitation	honeypots, honeypatches	Low-interaction and high-interaction honeypots, vulnerability database, game-theoretic engine	Exploitation time, re-configuration Cost, game utility	Assumes rational attackers, parameter heuristics
[137]	Proactive deception and attacker profiling in IoV	Remote hijacking, data breaches, unauthorized access (e.g., OTA injection, CAN bus attacks)	Digital Twin (DT)-based deception with blockchain integration	Internet/ Networking (decoys, baiting, breadcrumbs, firewalls), physical System, deception environment	Cost-effectiveness, adaptiveness to evolving threats, realism of deception, immutability of attacker logs/ TTPs (via blockchain)	No real-world testing, context-dependent applicability, scalability challenges with blockchain storage

filesystem manipulation and the MAVLink flightpath hijacking via QGroundControl were the basis for its evaluation. HoneyDrone once again came off well against the attackers through mirroring while recording all the interactions. Still, a possible drawback of being identified as a honeypot is the static signal characteristics, so the intention is to dynamically address this issue with the upcoming work on signal emulation and better scalability for concurrent connections.

In [140], the authors presented a scheme based on deep reinforcement learning (DRL), which serves as a deception strategy to successfully fight reactive jamming in IRS-assisted UAV communication systems. The system instead uses a UAV as an aerial base station in the air and trees the control element as the Intelligent Reflecting Surfaces (IRS) to connect the signal better through the controllable non-line-of-sight (NLOS) and direct (LOS) paths. The adversary in this case is a stealthy reactive jammer that selects the high-power transmissions

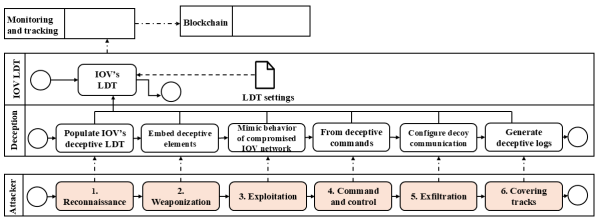

Fig. 18. The realization of DECEPTWIN during the remote access attack on a vehicle within IOV environment in [137].

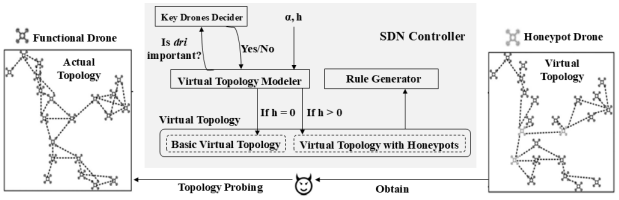

Fig. 19. The designed SDN-based topology deception scheme in [141].

after it has sensed the channel activity. To counter this, the framework includes a Deep Q-Network (DQN) that supports dynamic power allocation and IRS optimization (e.g., phase shifts, positioning), deceiving the jammer through a two-layer strategy: dummy packet transmission to profile the jammer's behavior and power-boosted decoy channels to misdirect attacks toward non-critical frequencies. The DQN agent learns an optimal policy via Q-learning, balancing energy efficiency and transmission rates. Simulations have shown the improvement in Total Received Power (TRP), the reduction of transmit power, and the enhancement of resilience against the baseline methods (e.g., Q-learning only or non-IRS systems). Apart from limitations, authors have made a single UAV/jammer, perfect environmental knowledge, and unexplored real-world deployment challenges (e.g., IRS mobility, multi-jammer scenarios).

In [141], the authors addressed an underexplored challenge of protecting key drones in UAV networks from targeted attacks. These drones, besides being tracers or service hosts, may be a means for some malicious elements to compromise the drone network. The authors begin with the attacker strategy analysis (which may include degree, betweenness, and closeness centrality), and the result is that even targeting a few key drones can cause significant performance degradation. To avert such a situation, they proposed an SDN-based topology deception scheme, which is made up of the following three parts: 1) a Key Drone Privider that decides by identifying critical drones with the help of a custom metric to quantify the connectivity loss; 2) a Virtual Topology Modeler that constructs the deceptive graphs by using the minimum spanning trees and the strategically placed honeypot drones; 3) a Rule Generator that makes recourse to the SDN controller by redirecting the probing attempts (e.g., via TTL-limited traceroute) to faked topologies. It is evident from the illustration in Fig. 19 that non-critical probes get a basic virtual topology, while the critical ones respond in a way that misleads the attackers, by guiding them to honeypots. The plan is executed in a Mininet-WiFi environment, controlled by a Ryu controller, and is notable for its high resilience as exhibited by statistics like Connectivity Loss (CL) and the success rate of attackers, which show even a small difference in honeypot deployment reduces attack impact. Nevertheless, the method presupposes the presence of all drones with OpenFlow support and runs into scalability problems when the network is larger.

In [142], the authors proposed a protection system for UAV networks that avails of honeypot-based deception mechanism and uses learning-based game theory to entice UAVs into sharing their Valid Defense Data (VDD) with the other UAVs. In the frame of this idea, every UAV carries low/ or medium-interaction honeypot modules that catch attack data, with a ground control station (GCS) designing VDD-reward contracts for UAVs to deal with issues like information asymmetry and UAVs' selfish behaviors. The contract is based on partial information asymmetry, and contract theory is utilized to ensure that the incentives are optimal, fair, and budget-feasible, with guaranteed theoretical feasibility and fairness for contracts. For scenarios where information is completely asymmetric, the framework uses a multi-agent Markov game model, and a two-tier Policy Hill-Climbing (PHC) reinforcement learning algorithm that is designed to optimize the strategies dynamically to the UAVs and GCS, enabling adaptive decision-making in the time-varying environments. As illustrated in Fig. 20, the system makes use of UAV private information like VDD volume and communication delays in contract menu, that is GCS is cutting the rewards based on the different UAV types the GCS has and is using Air-to-Air (A2A) and Air-to-Ground (A2G) links for collaborative defense. Simulation experiments have shown that the proposed method noticeably elevates the UAV utility, the rates of participation, and the defensive effectiveness compared to standard mechanisms. Nevertheless, the framework takes a GCS to be trusted yet it does not question the possibility of UAV collusive attacks. Improvements will focus on understanding the activities of hostile UAVs, combining active network conditions and seeing whether stealthy methods such as federated learning might also work and the implementation of a blockchain model on which the free of trust data exchange can take place to make the confidentiality and expansiveness even more secure.

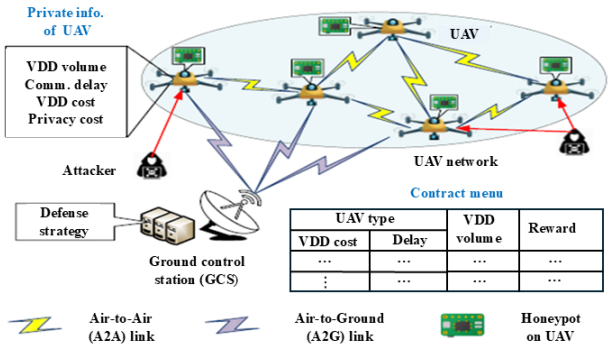

Fig. 20. The illustration of the incentive-driven honeypot game for collaborative defense in [142].

Table IX summarizes all the discussed works in unmanned aerial vehicles, providing a comparative overview of their defense goals, targeted attack types, deception methods, architectural components, evaluation metrics, and identified lim-

itations.

XIV. Open Issues and Key Insights

Despite significant progress in cyber deception research across diverse domains, several issues are still unresolved. The main reasons for this are the rising complexity of infrastructures, the changing patterns of attacker behavior, and the poor implementation of deception frameworks. This section reviews the main open questions accompanied by the most representative findings from the surveyed literature.

A. Incomplete or Simplistic Attacker Modeling

Deception strategies often have overly simplified attacker models, which usually include the assumptions of rational, single-agent, or static behaviors. Game-theoretic approaches in cloud settings, as in [91], are a good example as they usually take into account cost-sensitive attackers but fail to include adaptive adversaries. Advanced modeling is exemplified in [102] and [100], which discuss dynamic and context-aware behaviors in wireless networks. Moreover, adaptive attacker responses are included in [140], which augment the realism. However, there are very few systems that can mirror cognitive biases, collaborative threats, or deception-aware APTs. Cognitive, probabilistic, and multi-agent models should be used to represent the decision of the attacker under advanced uncertainty.

B. Limited Coverage of Post-compromise, Multi-Stage Attacks

Deception strategies mainly find their place in the primary attack phases, namely, scanning, probing, and initial access. Many of the attackers, however, tend to be open to the traps and they execute lateral movement or data exfiltration on their own without even being noticed. The deceiving attack approach is mostly implemented in the beginning phase in ICS and smart grids (for example, [114] and [118]). The same is true for IoT (for example, [129] and [130]), where early traffic analysis is the main focus. A limited number of systems include deception in the runtime workflows or administrative interface. The proposed solutions should cope with final phases of the attack kill chain using decoys designed for the post-compromise phases like privilege escalation and persistence.

C. Limited Adaptivity and Real-time Reconfiguration

The current systems are mostly characterized by static honeypots and inflexible deception logic. There are only a few exceptions, such as [116] and [140], which adapt their deception policies dynamically based on the feedback. The cloud environment has [96] its virtual decoy redeployment solution. Nonetheless, the majority of the systems do not have real-time environmental awareness and self-reconfiguration. It is the application of AI, mainly through reinforcement learning, behavior modeling, and predictive analytics, that is the key to flexible adaptation of deception strategies.

D. Scalability and Fidelity in Resource-constrained Environments

Trade-offs between scalability and deception realism are often mandated by resource constraints. One example is [105] where signaling games are employed for a reduction in energy consumption, whereas [139] and [136] get lightweight deception appropriate for mobile systems. In the cloud environment, [97] and [122] show dynamic orchestration of decoy under resource limitations. Combining low- and high-interaction techniques in a honeypot and dynamically changing to fit the perceived threat level are very good but come with the challenge of creating formal optimization frameworks.

E. Lack of Standardized Metrics and Quantification Frameworks

Current assessment methods vary across different fields. Metrics like engagement time, attacker delay, and resource cost are usually used separately. Some works, such as [122] in smart grids, introduce innovative privacy-oriented metrics (e.g., k-anonymity), while others like [114] and [123] apply reward-based metrics from game theory. Similarly, measuring deception success, attacker confusion, or resilience improvements is not achievable as long as there is no universally adaptable and context-aware framework. The immediate requirement is to normalize testing processes and set up reproducible benchmarks.

F. Limited Testbeds and Validation Platforms

Validation of the majority of deception systems is accomplished through simulation, which leads to the diminished credibility for their real-world deployment. Just like [107] [142] and [137], they put forward interesting ideas but do not demonstrate a physical testbed. Moreover, even in safety-critical areas such as smart grids that have no choice but to depend on simulated environments, there are risks of deployment. Modular scalable testbeds emulating multi-layer architecture and realistic user and attacker behavior are prerequisites for credible evaluation and operational trust.

G. Security and Privacy Trade-Offs in Collaborative Deception

Collaboration through deception, particularly in distributed systems (such as UAV swarms, IoV), brings forth coordination, trust, and privacy challenges. In the case of UAVs, [142] looks at role-switching and shared intelligence among drones. In IoV, [137] uses digital twins for proactive deception. Nevertheless, the majority of the systems don't address claims like adversarial control injection, secure communication, and privacy-preserving coordination. Countermeasures like federated learning, blockchain-based consensus, and zero-trust coordination models are still under-exploited in cyber deception.

H. Integration Gaps with Existing Defense Ecosystems

Deception platforms are frequently set up as isolated systems, which means they do not communicate with SIEM, IDS/IPS, and cyber threat intelligence (CTI) systems. Apart from the tools given, such as [114] or [129] that have shown effectiveness in the engagement of the attacker, they usually miss the mechanisms to put the ideas acquired through this process back into the general defense workflow systematically. The cloud-based models, such as [95], that suggest improved integration potential are left behind due to the unavailability of standardized APIs, logging formats, and data models that are

TABLE IX. SUMMARY OF CYBER DECEPTION TECHNIQUES IN UNMANNED AERIAL VEHICLES

Reference	Defense Goal	Attack Type	Deception Method	Architecture Components	Evaluation Metrics	Uncovered Limitations
[139]	Detect UAV attacks, gather threat intelligence, divert attackers from real drones	Wi-Fi hijacking, Telnet/SSH/FTP exploits, MAVLink protocol abuse	Medium-interaction honeypot	Network Interface Emulator (NIE), UAV emulation, File System Emulator, configuration file	Performance, effectiveness, realism	Static signal behavior, limited to emulated protocols
[140]	Mitigate jamming attacks in IRS-assisted UAV communications	Reactive jamming (intelligent, stealthy)	dummy packets, decoy channels	IRS with passive reflecting elements, UAV as aerial BS, DRL-driven power allocation, Q-learning and DQN for dynamic resource allocation	Rate of communication under jamming, transmit power efficiency, training time of DRL methods	Increased resource usage, assumes jammers attack highest power channels
[141]	Protect key drones in UAV networks from targeted attacks	Topology probing-based targeted attacks	Virtual topology generation (spanning trees) to hide real key drones, Honeypot drones	SDN controller, Key Drone Decider, Virtual Topology Modeler, Rule Generator, OpenFlow-enabled drones, Honeypot drones	Connectivity loss, normalized communication connectivity decrement, importance degree from attacker's view, optimal transmit power and resource usage	Assumes OpenFlow-enabled drones, scalability issues
[142]	Protect UAV networks by incentivizing collaborative defense via honeypot data sharing	Cyber threats (e.g., DoS, hijacking, data theft), free-riding attacks	Contract theory + multi-agent learning for honeypot sharing	Low/ medium-interaction honeypots, Ground Control Station (GCS), UAVs	Connectivity loss, social surplus, defense effectiveness, UAV/GCS utilities, Attack detection rate, CPU utilization of honeypots	Scalability challenges

valid for interoperability. The operational impact of deception systems will be maximized only when they are designed with interoperability as a guiding principle. The introduction of a common and standardized framework, like STIX in the CTI, makes it possible to create actionable deception-derived feedback that could be accessed and used by various other security solutions.

I. Emerging Technologies as Catalysts for Next-Gen Deception

Digital twins, large language models (LLMs), and multi-agent AI have a vast potential for the development of cyber deception. In the IoT sector, dynamic flow redirection has been put forward by [69] as the way to mislead the intruders. Digital twins have been used in the process of anticipatory deception in [137], which provided an opportunity to the users for simulating and predicting the attacker's behavior. Likewise, in UAV networks [140] and smart grids [122], AI-driven and privacy-preserving deception architectures have also been the focus of research. Yet, these things remain primarily theoretical or applicable to particular areas only. To be queried within a full range, these key players should be utilized in a cross-domain, full-stack deception model. As an illustration, digital twins can be used to replicate real environments, thus allowing the defenders to either emulate the vulnerable system in the physical absence of assets or simulate the real-world scenarios with the data of actual threats. LLMs can be incorporated into generating deceptive responses, analyzing adversarial traces, coordinating multi-agent deception strategies, and determining adaptive actions to disrupt, tolerate, or mislead attacks. Joining these two technologies together can greatly modify the flexibility, realism, and intelligence of the deception systems in various functional situations.

J. Challenges in Realism and Fingerprint Resistance

Over time, as the decoys have become more sophisticated, adversaries have also advanced when it comes to the detection of such decoys by means of timing anomalies, low-level operating system artifacts, or service inconsistencies. Researches like [110] and [99] show that the mid-interaction honeypots can be identified by their fingerprints. The presence of behavioral mimicry, dynamic response crafting, and protocol emulation with the help of artificial intelligence makes it possible to stay one step ahead of the attacker reconnaissance. In most systems, these capabilities are still not utilized adequately.

XV. CONCLUSION

This survey sorts the cyber deception techniques in a comparative manner for different domains like CE, WN, CPS, ICS, SG, IOT, IOV, and UAV. The study rigorously studied and communicated the architectural structures, attack types, deception techniques, evaluation metrics, and limitations present in each domain. The findings, which are collected and represented in the unified tables, pave the way for an easy comparison. Besides, the survey deals with the most frequently met challenges and the research directions that remain open, which gives practical insights and support in the design of adaptive and effective deception-based defenses tailored to different operational environments. In our future work, we intend to overcome some of the limitations discussed in the previous section, and our focus will be on utilizing Large Language Models (LLMs) for cyber deception. We aim to examine the efficacy of the LLMs in the generation of deceptive responses, the interpretation of adversarial behavior, and the decision-making of adaptive strategies for disruption, misinformation, or acceptance of cyber threats.

ACKNOWLEDGMENT

This project is carried out under the MOBIDOC scheme, funded by the Ministry of Higher Education and Scientific Research through the PromEssE project and managed by the ANPR.

REFERENCES

[1] L. Zhang and V. L. Thing, "Three decades of deception techniques in active cyber defense-retrospect and outlook," *Computers & Security*, vol. 106, p. 102288, 2021.

[2] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–36, 2018.

[3] J. Pawlick, E. Colbert, and Q. Zhu, "A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–28, 2019.

[4] M. Zhu, A. H. Anwar, Z. Wan, J.-H. Cho, C. A. Kamhoua, and M. P. Singh, "A survey of defensive deception: Approaches using game theory and machine learning," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2460–2493, 2021.

[5] M. A. Sayed, M. Rahman, M. A. I. Khan, and D. Tosh, "A survey of network requirements for enabling effective cyber deception," *arXiv preprint arXiv:2309.00184*, 2023.

[6] X. Qin, F. Jiang, M. Cen, and R. Doss, "Hybrid cyber defense strategies using honey-x: A survey," *Computer Networks*, vol. 230, p. 109776, 2023.

[7] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A comprehensive survey on cyber deception techniques to improve honeypot performance," *Computers & Security*, p. 103792, 2024.

[8] M. Johnson and J. Meyeraan, "Military deception: Hiding the real-showing the fake," *USAF Joint Forces Staff College, Joint and Combined Warfighting School*, vol. 7, 2003.

[9] M. H. Almeshekah and E. H. Spafford, "Cyber security deception," *Cyber Deception: Building the Scientific Foundation*, pp. 23–50, 2016.

[10] B. Whaley, "Toward a general theory of deception," *The Journal of Strategic Studies*, vol. 5, no. 1, pp. 178–192, 1982.

[11] C. Stoll, "The cuckoo's egg: Tracing a spy through the maze of computer espionage, 1989," *Google Scholar Google Scholar Digital Library Digital Library*.

[12] L. Spitzner, "The honeynet project: Trapping the hackers," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 15–23, 2003.

[13] N. Provos *et al.*, "A virtual honeypot framework." in *USENIX Security Symposium*, vol. 173, no. 2004, 2004, pp. 1–14.

[14] M. H. Almeshekah and E. H. Spafford, "Planning and integrating deception into computer security defenses," in *Proceedings of the 2014 New Security Paradigms Workshop*, 2014, pp. 127–138.

[15] A. Aly, M. Fayez, M. M. Al-Qutt, and A. Hamad, "Navigating the deception stack: In-depth analysis and application of comprehensive cyber defense solutions," *International Journal of Intelligent Computing and Information Sciences*, vol. 23, no. 4, pp. 50–65, 2023.

[16] S. T. Trassare, R. Beverly, and D. Alderson, "A technique for network topology deception," in *MILCOM 2013-2013 IEEE Military Communications Conference*. IEEE, 2013, pp. 1795–1800.

[17] S. Achleitner, T. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, "Cyber deception: Virtual networks to defend insider reconnaissance," in *Proceedings of the 8th ACM CCS international workshop on managing insider security threats*, 2016, pp. 57–68.

[18] B. Wang and B. Lu, "A network deception defense mechanism based on virtual topology generation," in *International Conference on Intelligent Systems, Communications, and Computer Networks (ISCCN 2023)*, vol. 12702. SPIE, 2023, pp. 641–647.

[19] A. Clark, K. Sun, and R. Poovendran, "Effectiveness of ip address randomization in decoy-based moving target defense," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 678–685.

[20] A. R. Chavez, W. M. Stout, and S. Peisert, "Techniques for the dynamic randomization of network attributes," in *2015 international carnahan conference on security technology (ICCST)*. IEEE, 2015, pp. 1–6.

[21] J. Narantuya, S. Yoon, H. Lim, J.-H. Cho, D. S. Kim, T. Moore, and F. Nelson, "Sdn-based ip shuffling moving target defense with multiple sdn controllers," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S)*. IEEE, 2019, pp. 15–16.

[22] L. Alt, R. Beverly, and A. Dainotti, "Uncovering network tarpits with degreaser," in *Proceedings of the 30th Annual Computer Security Applications Conference*, 2014, pp. 156–165.

[23] L. Shing, "An improved tarpit for network deception," Ph.D. dissertation, Monterey, California: Naval Postgraduate School, 2016.

[24] M. A. Sayed, A. H. Anwar, C. Kiekintveld, and C. Kamhoua, "Honeypot allocation for cyber deception in dynamic tactical networks: A game theoretic approach," in *International Conference on Decision and Game Theory for Security*. Springer, 2023, pp. 195–214.

[25] Z. Morić, V. Dakić, and D. Regvart, "Advancing cybersecurity with honeypots and deception strategies," in *Informatics*, vol. 12, no. 1. MDPI AG, 2025, p. 14.

[26] L. Patil, S. Desai, and A. Singh, "Honeypot based secure network system," 2021.

[27] X. Yang, J. Yuan, H. Yang, Y. Kong, H. Zhang, and J. Zhao, "A highly interactive honeypot-based approach to network threat management," *Future Internet*, vol. 15, no. 4, p. 127, 2023.

[28] D. Fraunholz, D. Krohmer, F. Pohl, and H. D. Schotten, "On the detection and handling of security incidents and perimeter breaches-a modular and flexible honeytoken based framework," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2018, pp. 1–4.

[29] E. Filippi, "Honeyport-a scalable meta-honeypot system for security applications," Ph.D. dissertation, Politecnico di Torino, 2019.

[30] F. Araujo, K. W. Hamlen, S. Biedermann, and S. Katzenbeisser, "From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 942–953.

[31] F. Araujo, M. Shapouri, S. Pandey, and K. Hamlen, "Experiences with {Honey-Patching} in active cyber security education," in *8th Workshop on Cyber Security Experimentation and Test (CSET 15)*, 2015.

[32] J. Avery and E. H. Spafford, "Ghost patches: Fake patches for fake vulnerabilities," in *ICT Systems Security and Privacy Protection: 32nd IFIP TC 11 International Conference, SEC 2017, Rome, Italy, May 29-31, 2017, Proceedings 32*. Springer, 2017, pp. 399–412.

[33] J. K. Avery, "The application of deception to software security patching," Ph.D. dissertation, Purdue University, 2017.

[34] M. Albanese, E. Battista, and S. Jajodia, "A deception based approach for defeating os and service fingerprinting," in *2015 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2015, pp. 317–325.

[35] M. A. Rahman, M. M. Hasan, M. H. Manshaei, and E. Al-Shaer, "A game-theoretic analysis to defend against remote operating system fingerprinting," *Journal of Information Security and Applications*, vol. 52, p. 102456, 2020.

[36] N. C. Rowe, J. Rrushi *et al.*, *Introduction to cyberdeception*. Springer, 2016.

[37] F. Araujo, *Engineering cyber-deceptive software*. The University of Texas at Dallas, 2016.

[38] F. Araujo, W. Kevin *et al.*, "Compiler-instrumented, dynamic {Secret-Redaction} of legacy processes for attacker deception," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 145–159.

[39] P. Kaghazgaran and H. Takabi, "Toward an insider threat detection framework using honey permissions." *J. Internet Serv. Inf. Secur.*, vol. 5, no. 3, pp. 19–36, 2015.

[40] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th annual computer security applications conference*, 2010, pp. 1–9.

[41] N. Virvilis, B. Vanautgaerden, and O. S. Serrano, "Changing the game: The art of deceiving sophisticated attackers," in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*. IEEE, 2014, pp. 87–97.

[42] P. Peng, C. Xu, L. Quinn, H. Hu, B. Viswanath, and G. Wang, "What happens after you leak your password: Understanding credential sharing on phishing sites," in *Proceedings of the 2019 ACM Asia conference on computer and communications security*, 2019, pp. 181–192.

[43] C. Katsinis and B. Kumar, "A framework for intrusion deception on web servers," in *2013 International Conference on Internet Computing, ICOMP'13*, 2013.

[44] D. P. Julian, "Delaying-type responses for use by software decoys," Ph.D. dissertation, Monterey, California. Naval Postgraduate School, 2002.

[45] S. Sugrim, S. Venkatesan, J. A. Youzwak, C.-Y. J. Chiang, R. Chadha, M. Albanese, and H. Cam, "Measuring the effectiveness of network deception," in *2018 IEEE international conference on Intelligence and Security Informatics (ISI)*. IEEE, 2018, pp. 142–147.

[46] M. M. Islam and E. Al-Shaer, "Active deception framework: An extensible development environment for adaptive cyber deception," in *2020 IEEE Secure Development (SecDev)*. IEEE, 2020, pp. 41–48.

[47] D. Gavrilis, I. Chatzis, and E. Dermatas, "Flash crowd detection using decoy hyperlinks," in *2007 IEEE International Conference on Networking, Sensing and Control*. IEEE, 2007, pp. 466–470.

[48] D. Brewer, K. Li, L. Ramaswamy, and C. Pu, "A link obfuscation service to detect webbots," in *2010 IEEE International Conference on Services Computing*. IEEE, 2010, pp. 433–440.

[49] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents," in *Security and Privacy in Communication Networks: 5th International ICST Conference, SecureComm 2009, Athens, Greece, September 14-18, 2009, Revised Selected Papers 5*. Springer, 2009, pp. 51–70.

[50] C. M. McRae and R. B. Vaughn, "Phighting the phisher: Using web bugs and honeytokens to investigate the source of phishing attacks," in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*. IEEE, 2007, pp. 270c–270c.

[51] A. R. Petrunić, "Honeytokens as active defense," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2015, pp. 1313–1317.

[52] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici, "Honeygen: An automated honeytokens generator," in *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2011, pp. 131–136.

[53] M. Msaad, S. Srinivasa, M. M. Andersen, D. H. Audran, C. U. Orji, and E. Vasilomanolakis, "Honeysweeper: Towards stealthy honeytoken fingerprinting techniques," in *Nordic Conference on Secure IT Systems*. Springer, 2022, pp. 101–119.

[54] M. Kahlhofer, S. Achleitner, S. Rass, and R. Mayrhofer, "Honeyquest: Rapidly measuring the enticingness of cyber deception techniques with code-based questionnaires," in *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, 2024, pp. 317–336.

[55] J. Yuill, M. Zappe, D. Denning, and F. Feer, "Honeyfiles: deceptive files for intrusion detection," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.* IEEE, 2004, pp. 116–122.

[56] J. Voris, J. Jermyn, N. Boggs, and S. Stolfo, "Fox in the trap: Thwarting masqueraders via automated decoy document deployment," in *Proceedings of the eighth European workshop on system security*, 2015, pp. 1–7.

[57] M. Lazarov, J. Onaolapo, and G. Stringhini, "Honey sheets: What happens to leaked google spreadsheets?" in *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*, 2016.

[58] R. C. Timmer, D. Liebowitz, S. Nepal, and S. Kanhere, "Tsm: Measuring the enticement of honeyfiles with natural language processing," *arXiv preprint arXiv:2203.07580*, 2022.

[59] M. H. Almeshekah, "Using deception to enhance security: A taxonomy, model, and novel uses," Ph.D. dissertation, Purdue University, 2015.

[60] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, "Kamouflage: Loss-resistant password management," in *Computer Security–ESORICS 2010: 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings 15*. Springer, 2010, pp. 286–302.

[61] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 145–160.

[62] J. Dani, B. McCulloh, and N. Saxena, "When ai defeats password deception! a deep learning framework to distinguish passwords and honeywords," *arXiv preprint arXiv:2407.16964*, 2024.

[63] A. Juels and T. Ristenpart, "Honey encryption: Security beyond the brute-force bound," in *Advances in Cryptology–EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*. Springer, 2014, pp. 293–310.

[64] O. T. Taofeek, M. Alawida, A. Alabdulatif, A. E. Omolara, and O. I. Abiodun, "A cognitive deception model for generating fake documents to curb data exfiltration in networks during cyber-attacks," *IEEE Access*, vol. 10, pp. 41 457–41 476, 2022.

[65] Y. Hu, Y. Lin, E. S. Parolin, L. Khan, and K. Hamlen, "Controllable fake document infilling for cyber deception," *arXiv preprint arXiv:2210.09917*, 2022.

[66] Y. Park and S. J. Stolfo, "Software decoys for insider threat," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 2012, pp. 93–94.

[67] M. Ge, J.-H. Cho, D. Kim, G. Dixit, and I.-R. Chen, "Proactive defense for internet-of-things: moving target defense with cyberdeception," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 1, pp. 1–31, 2021.

[68] S. Wang, Q. Pei, Y. Zhang, X. Liu, and G. Tang, "A hybrid cyber defense mechanism to mitigate the persistent scan and foothold attack," *Security and Communication Networks*, vol. 2020, no. 1, p. 8882200, 2020.

[69] Y. Zhou, G. Cheng, and S. Yu, "An sdn-enabled proactive defense framework for ddos mitigation in iot networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5366–5380, 2021.

[70] S. Wang, Q. Pei, J. Wang, G. Tang, Y. Zhang, and X. Liu, "An intelligent deployment policy for deception resources based on reinforcement learning," *IEEE Access*, vol. 8, pp. 35 792–35 804, 2020.

[71] T. Kong, L. Wang, D. Ma, Z. Xu, Q. Yang, Z. Lu, and Y. Lu, "Automated honeynet deployment strategy for active defense in container-based cloud," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2020, pp. 483–490.

[72] N. C. Abay, C. G. Akcora, Y. Zhou, M. Kantarcioglu, and B. Thuraisingham, "Using deep learning to generate relational honeydata," *Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings*, pp. 3–19, 2019.

[73] A. Basak, C. Kamhoua, S. Venkatesan, M. Gutierrez, A. H. Anwar, and C. Kiekintveld, "Identifying stealthy attackers in a game theoretic framework using deception," in *Decision and Game Theory for Security: 10th International Conference, GameSec 2019, Stockholm, Sweden, October 30–November 1, 2019, Proceedings 10*. Springer, 2019, pp. 21–32.

[74] O. Thakoor, M. Tambe, P. Vayanos, H. Xu, C. Kiekintveld, and F. Fang, "Cyber camouflage games for strategic deception," in *Decision and Game Theory for Security: 10th International Conference, GameSec 2019, Stockholm, Sweden, October 30–November 1, 2019, Proceedings 10*. Springer, 2019, pp. 525–541.

[75] A. H. Anwar, C. Kamhoua, and N. Leslie, "Honeypot allocation over attack graphs in cyber deception games," in *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 502–506.

[76] Mitre att&ck. Accessed on 03-01-2025. [Online]. Available: https://attack.mitre.org/

[77] M. Zambianco, C. Facchinetti, and D. Siracusa, "A proactive decoy selection scheme for cyber deception using mitre att&ck," *arXiv preprint arXiv:2404.12783*, 2024.

[78] A. Sayari, Y. Djemaiel, S. Rekhis, A. Mabrouk, and B. Jerbi, "Attack modeling and cyber deception resources deployment using multi-layer graph," in *International Conference on Advanced Information Networking and Applications*. Springer, 2022, pp. 560–572.

[79] A. Sayari, S. Rekhis, Y. Djemaiel, A. Mabrouk, and W. Mahouachi, "Decept-cti: A framework for enhancing cyber deception strategies through nlp-based extraction of cti from unstructured reports," in *2024 22nd International Symposium on Network Computing and Applications (NCA)*. IEEE, 2024, pp. 286–293.

[80] Mitre d3fend. Accessed on 17-01-2025. [Online]. Available: https://d3fend.mitre.org/

[81] Mitre engage. Accessed on 17-03-2025. [Online]. Available: https://engage.mitre.org/matrix/?phase=operate

[82] A. H. Anwar and C. A. Kamhoua, "Cyber deception using honeypot allocation and diversity: A game theoretic approach," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 543–549.

[83] T. Zhang, L. Huang, J. Pawlick, and Q. Zhu, "Game-theoretic analysis of cyber deception: Evidence-based strategies and dynamic risk mitigation," *Modeling and Design of secure Internet of Things*, pp. 27–58, 2020.

[84] Y.-T. Yang and Q. Zhu, "Game-theoretic foundations for cyber resilience against deceptive information attacks in intelligent transportation systems," *arXiv preprint arXiv:2412.04627*, 2024.

[85] A. Sayari, S. Rekhis, Y. Djemaiel, and A. Mabrouk, "An adaptive reinforcement learning-based approach for effective cyber denial and deception strategies finding," in *International Conference on Advanced Information Networking and Applications*. Springer, 2025, pp. 224–234.

[86] A. Charpentier, N. Boulahia Cuppens, F. Cuppens, and R. Yaich, "Deep reinforcement learning-based defense strategy selection," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–11.

[87] Y. Zhang, F. Liu, and H. Chen, "Optimal strategy selection for cyber deception via deep reinforcement learning," in *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*. IEEE, 2022, pp. 1841–1847.

[88] A. Sayari, S. Rekhis, and A. Mabrouk, "A game-theoretic approach with decoy qubits for quantum superdense coding security," in *2024 International Conference on Ubiquitous Networking (UNet)*, vol. 10. IEEE, 2024, pp. 1–8.

[89] U. A. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. T. Alharbi, and N. Albaqami, "Cloud security threats and solutions: A survey," *Wireless Personal Communications*, vol. 128, no. 1, pp. 387–413, 2023.

[90] V. Mahajan and S. K. Peddoju, "Integration of network intrusion detection systems and honeypot networks for cloud security," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, pp. 829–834.

[91] M. T. Adili, A. Mohammadi, M. H. Manshaei, and M. A. Rahman, "A cost-effective security management for clouds: A game-theoretic deception mechanism," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 98–106.

[92] A. Aydeger, N. Saputro, and K. Akkaya, "Cloud-based deception against network reconnaissance attacks using sdn and nfv," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 279–285.

[93] K. D. Singh, "Securing of cloud infrastructure using enterprise honeypot," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. IEEE, 2021, pp. 1388–1393.

[94] H. Li, Y. Guo, S. Huo, H. Hu, and P. Sun, "Defensive deception framework against reconnaissance attacks in the cloud with deep reinforcement learning," *Science China Information Sciences*, vol. 65, no. 7, p. 170305, 2022.

[95] A. El-Kosairy and N. Abdelbaki, "Deception as a service: intrusion and ransomware detection system for cloud computing (irds4c)," *Advances in Computational Intelligence*, vol. 3, no. 3, p. 9, 2023.

[96] V. D. Priya and S. S. Chakkaravarthy, "Containerized cloud-based honeypot deception for tracking attackers," *Scientific Reports*, vol. 13, no. 1, p. 1437, 2023.

[97] M. Zambianco, C. Facchinetti, R. Doriguzzi-Corin, and D. Siracusa, "Resource-aware cyber deception for microservice-based applications," *IEEE Transactions on Services Computing*, 2024.

[98] R. Nazir, A. A. Laghari, K. Kumar, S. David, and M. Ali, "Survey on wireless network security," *Archives of Computational Methods in Engineering*, pp. 1–20, 2021.

[99] D. B. Rawat, N. Sapavath, and M. Song, "Performance evaluation of deception system for deceiving cyber adversaries in adaptive virtualized wireless networks," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 401–406.

[100] A. Adebayo and D. B. Rawat, "Deceptor-in-the-middle (ditm): cyber deception for security in wireless network virtualization," in *2020 IEEE 17th annual consumer communications & networking conference (CCNC)*. IEEE, 2020, pp. 1–6.

[101] A. A. Adebayo and D. B. Rawat, "Cyber deception for wireless network virtualization using stackelberg game theory," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021, pp. 1–6.

[102] S. Nan, S. Brahma, C. A. Kamhoua, and N. O. Leslie, "Mitigation of jamming attacks via deception," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–6.

[103] Q. He, E. Yang, S. Fang, and S. Zhao, "Revisiting wireless breath and crowd inference attacks with defensive deception," *IEEE/ACM Transactions on Networking*, 2024.

[104] I. C. K. Sihomnou, A. Benslimane, A. H. A. Hemida, G. Deugoue, C. Kamhoua, and C. So-In, "Mitigating energy attacks in wireless sensor networks using deception: A game theoretic approach," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 4792–4797.

[105] I. C. K. Sihomnou, A. Benslimane, A. H. Anwar, G. Deugoue, and C. Kamhoua, "Cyber deception against battery drain dos attacks in wireless sensor networks using signaling game," *IEEE Access*, 2025.

[106] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.

[107] M. O. Sayin and T. Başar, "Deception-as-defense framework for cyber-physical systems," in *Safety, Security and Privacy for Cyber-Physical Systems*. Springer, 2021, pp. 287–317.

[108] W. Tian, X. Ji, W. Liu, G. Liu, R. Lin, J. Zhai, and Y. Dai, "Defense strategies against network attacks in cyber-physical systems with analysis cost constraint based on honeypot game model." *Computers, Materials & Continua*, vol. 60, no. 1, 2019.

[109] G. K. Edwin, S. V. Ewards, G. J. W. Kathrine, G. M. Palmer, A. Bertia, and S. Vijay, "Honeypot based intrusion detection system for cyber physical system," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. IEEE, 2022, pp. 958–962.

[110] Y. Sun, Z. Tian, M. Li, S. Su, X. Du, and M. Guizani, "Honeypot identification in softwarized industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5542–5551, 2020.

[111] P. C. Kamdem, A. Zemkoho, L. Njilla, M. Nkenlifack, and C. Kamhoua, "Two-layer deception model based on signaling games against cyber attacks on cyber-physical systems." *IEEE Access*, 2024.

[112] M. Nankya, R. Chataut, and R. Akl, "Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies," *Sensors*, vol. 23, no. 21, p. 8840, 2023.

[113] M. Winn, M. Rice, S. Dunlap, J. Lopez, and B. Mullins, "Constructing cost-effective and targetable industrial control system honeypots for production networks," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 47–58, 2015.

[114] E. López-Morales, C. Rubio-Medrano, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 279–291.

[115] T. Machida, D. Yamamoto, Y. Unno, and H. Kojima, "Novel deception techniques for malware detection on industrial control systems," *Journal of Information Processing*, vol. 29, pp. 559–571, 2021.

[116] J. Luo, T. Liu, M. Liang, and N. Hu, "A hmm-based ics adaptive deception defense framework," in *2023 8th International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2023, pp. 359–366.

[117] B. Paul, A. Sarker, S. H. Abhi, S. K. Das, M. F. Ali, M. M. Islam, M. R. Islam, S. I. Moyeen, M. F. R. Badal, M. H. Ahamed *et al.*, "Potential smart grid vulnerabilities to cyber attacks: Current threats and existing mitigation strategies," *Heliyon*, vol. 10, no. 19, 2024.

[118] K. Wang, M. Du, S. Maharjan, and Y. Sun, "Strategic honeypot game model for distributed denial of service attacks in the smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2474–2482, 2017.

[119] C. Dalamagkas, P. Sarigiannidis, D. Ioannidis, E. Iturbe, O. Nikolis, F. Ramos, E. Rios, A. Sarigiannidis, and D. Tzovaras, "A survey on honeypots, honeynets and their applications on smart grid," in *2019 IEEE conference on network softwarization (NetSoft)*. IEEE, 2019, pp. 93–100.

[120] D. Mashima, Y. Li, and B. Chen, "Who's scanning our smart grid? empirical study on honeypot data," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[121] D. Mashima, D. Kok, W. Lin, M. Hazwan, and A. Cheng, "On design and enhancement of smart grid honeypot system for practical collection of threat intelligence," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*, 2020.

[122] D. Yang, D. Mashima, W. Lin, and J. Zhou, "Decied: Scalable k-anonymous deception for iec61850-compliant smart grid systems," in *Proceedings of the 6th ACM on Cyber-Physical System Security Workshop*, 2020, pp. 54–65.

[123] B. Li, Y. Shi, Q. Kong, C. Zhai, and Y. Ouyang, "Honeypot-enabled optimal defense strategy selection for smart grids," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[124] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEe Access*, vol. 7, pp. 82 721–82 743, 2019.

[125] F. Dang, Z. Li, Y. Liu, E. Zhai, Q. A. Chen, T. Xu, Y. Chen, and J. Yang, "Understanding fileless attacks on linux-based iot devices with honeycloud," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 482–493.

[126] B. Lingenfelter, I. Vakilinia, and S. Sengupta, "Analyzing variation among iot botnets using medium interaction honeypots," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0761–0767.

[127] W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An iot honeynet based on multiport honeypots for capturing iot attacks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3991–3999, 2019.

[128] O. Tsemogne, Y. Hayel, C. Kamhoua, and G. Deugoué, "Game-theoretic modeling of cyber deception against epidemic botnets in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2678–2687, 2021.

[129] M. S. Pour, J. Khoury, and E. Bou-Harb, "Honeycomb: A darknet-centric proactive deception technique for curating iot malware forensic artifacts," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[130] J. Bao, M. Kantarcioglu, Y. Vorobeychik, and C. Kamhoua, "Iot-flowgenerator: Crafting synthetic iot device traffic flows for cyber deception," *arXiv preprint arXiv:2305.00925*, 2023.

[131] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "Architecture, protocols, and security in iov: Taxonomy, analysis, challenges, and solutions," *Security and Communication Networks*, vol. 2022, no. 1, p. 1131479, 2022.

[132] S. Abbes and S. Rekhis, "Reinforcement learning-based virtual sensors provision in internet of vehicles (iov)," in *2022 IEEE 21st International Symposium on Network Computing and Applications (NCA)*, vol. 21. IEEE, 2022, pp. 217–224.

[133] A. Slim and S. Rekhis, "Reinforcement learning for intelligent sensor virtualization and provisioning in internet of vehicles (iov)," *IEEE Access*, 2024.

[134] M. Rajesh Babu and G. Usha, "A novel honeypot based detection and isolation approach (nhbadi) to detect and isolate black hole attacks in manet," *Wireless Personal Communications*, vol. 90, pp. 831–845, 2016.

[135] S. Sharma and A. Kaul, "A survey on intrusion detection systems and honeypot based proactive security mechanisms in vanets and vanet cloud," *Vehicular communications*, vol. 12, pp. 138–164, 2018.

[136] S. Panda, S. Rass, S. Moschoyiannis, K. Liang, G. Loukas, and E. Panaousis, "Honeycar: a framework to configure honeypot vulnerabilities on the internet of vehicles," *IEEE Access*, vol. 10, pp. 104 671–104 685, 2022.

[137] M. Iqbal, S. Suhail, and R. Matulevicius, "Deceptwin: Proactive security approach for iov by leveraging deception-based digital twins and blockchain," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, 2024, pp. 1–11.

[138] Z. Wang, Y. Li, S. Wu, Y. Zhou, L. Yang, Y. Xu, T. Zhang, and Q. Pan, "A survey on cybersecurity attacks and defenses for unmanned aerial systems," *Journal of Systems Architecture*, vol. 138, p. 102870, 2023.

[139] J. Daubert, D. Boopalan, M. Mühlhäuser, and E. Vasilomanolakis, "Honeydrone: A medium-interaction unmanned aerial vehicle honeypot," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–6.

[140] F. O. Olowononi, D. B. Rawat, C. A. Kamhoua, and B. M. Sadler, "Deep reinforcement learning for deception in irs-assisted uav communications," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE, 2022, pp. 763–768.

[141] Y. Tan, J. Liu, and J. Wang, "How to protect key drones in unmanned aerial vehicle networks? an sdn-based topology deception scheme," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 13 320–13 331, 2022.

[142] Y. Wang, Z. Su, A. Benslimane, Q. Xu, M. Dai, and R. Li, "Collaborative honeypot defense in uav networks: A learning-based game approach," *IEEE Transactions on Information Forensics and Security*, 2023.