Adversarial Exploitation of Emergent Dynamics in Smart Cities

Vahid Behzadan* and Arslan Munir†
Department of Computer Science
Kansas State University
Email: *behzadan@ksu.edu and †amunir@ksu.edu

***Abstract*—We investigate the paradigm of adversarial attacks that target the emergent dynamics of Complex Adaptive Smart Cities (CASCs). To facilitate the analysis of such attacks, we develop quantitative definitions and metrics of attack, vulnerability, and resilience in the context of CASC security. Furthermore, we propose multiple schemes for classification of attack surfaces and vectors in CASC, complemented with examples of practical attacks. Building on this foundation, we propose a framework based on reinforcement learning for simulation and analysis of attacks on CASC, and demonstrate its performance through two real-world case studies of targeting power grids and traffic management systems. We also remark on future research directions in analysis and design of secure smart cities and complex adaptive systems.**

***Keywords*—Smart Cities, Complex Systems, Resilience, Threat Modeling, Self-Organization, Emergent Behavior**

I. INTRODUCTION

The paradigm of smart city refers to the range of Information and Communication technologies (ICT) integrated with urban systems to enhance the efficiency and effectiveness of monitoring, management, and control of city operations [1]. With the rapid growth of cities [2] and the consequent rise of complexity in their management, it is widely believed that maintaining the sustenance and resilience of future cities will require pervasive and large-scale adoption of smart city technologies. Thus, recent years has witnessed accelerating advancements of such technologies in various direction that include Internet of Things (IoT), cloud and fog computing, smart grids, intelligent transportation systems, and many more [3].

Such technological developments in urban management are growingly deployed in many critical sectors, and will soon be deeply affecting the day to day activities of citizens. Hence, ensuring the security and resilience of smart cities is of paramount importance. Accordingly, a number of studies have aimed at identifying and classifying the vulnerabilities within various components of smart cities (e.g., [4]). Yet, the bulk of such studies are mainly focused on first-order vulnerabilities – that is, those that arise directly from arbitrarily reduced subsets of the smart city ecosystem and allow for immediate disruptions. Therefore, these studies fail to capture the threats rooted in the emergent behavior of smart city technologies as a whole. Furthermore, the body of research on the resilience and sustainability of smart cities (e.g., [5]) is generally concentrated around natural and unintentional causes of disruption, hence leaving the security perspective uncovered.

While understanding first-order vulnerabilities of smart technologies implanted into city operations is vital, it is also necessary to consider those that emerge from the complex dynamics and interactions of urban operations and systems. A prominent approach to the analysis of such dynamics is to model cities as instances of Complex Adaptive Systems (CAS) [5], which are characterized by complex behaviors that are the emergent results of nonlinear interactions between a large number of components at different levels of system's organization [6]. CAS are generally decentralized and governed by adaptive dynamics that enable their intrinsic adaptation and evolution in changing environments.

Such characteristics are also inherited by many of the technological components within the smart city paradigm: the decentralized and adaptive operation of CAS has prevailed in numerous engineering solutions for distributed system architectures, such as smart power grids [7], autonomous navigation [8], and IoT [9]. The CAS-based mechanisms of such distributed systems is indeed a promising approach to the challenging task of control and management of the increasingly complex and heterogeneous smart cities. In particular, the *Self-organization* aspect of CAS enables the emergence of order and pattern from uncoordinated actions of autonomous agents in multi-agent distributed settings [10].

While the distribution of functionalities and capabilities among multiple agents in CAS seemingly relieves the threats posed by single points of failure, the complexity of dynamics in such systems gives rise to unique challenges in quantifying and ensuring their resilience and robustness in hostile environments and adversarial conditions. The body of work on the sustainability of smart cities presents many contributions towards analysis of resilience against direct (i.e., first-order) perturbations, current state of the art leaves major gaps in understanding and enhancement of resilience against adversarial actions that target the higher-order dynamics of smart cities.

This paper aims to develop a consistent and quantitative approach towards analysis and enhancement of resilience in Complex Adaptive Smart Cities (CASCs) against adversarial actions. Accordingly, the main contributions of this paper are as follows: (1) We propose quantitative definitions of attack, vulnerability, and resilience in the context of CASC security. (2) We develop multiple schemes for classification of attack surfaces in CASC, and discuss generic instances of active and passive adversarial actions targeting these surfaces. (3) We propose a framework based on reinforcement learning for simulation and analysis of attacks on CASC. (4) We demonstrate the practical application of our proposed framework in 2 case studies: induction of cascade failures in power grids and disruption of traffic flow.

The remainder of this paper is organized as follows: Section

II provides an overview of CAS and the relevant background. Section III details our proposed definitions of attack, vulnerability, and resilience. Section IV presents classifications of vulnerabilities and attack surfaces in CASC, followed by the proposal of a framework for simulation of adversarial actions and analysis of their impact on CASC in Section V. Section VI demonstrates the application of this framework in two practical case studies. Finally, Section VII concludes the paper.

II. Background

In this section, we briefly introduce the paradigm of complex systems and their characteristics to provide the reader with an overview of fundamental concepts and notions required for the remainder of this paper. It must be noted that this background is by no means comprehensive, and the interested reader may refer to sources such as [11] for in-depth introductions to CAS.

A. Complex Adaptive Systems

Complexity, as a quantifiable measure, is yet to obtain a unified and consistent definition. From the multitude of definitions that have emerged from the field of complexity science [12], we abide by the definition presented by Mitchell [6]: "A complex adaptive system is a system in which large networks of components with simple rules of operation and no central control give rise to complex collective behavior, sophisticated information processing, and adaptation. Such systems exhibit nontrivial emergent and self-organizing behaviors."

Accordingly, the most general characteristics of CAS are identified as: (1) Large numbers of constituent elements and interactions; (2) Non-decomposability, i.e., components cannot be separately studied due to interactions; (3) Nonlinearity of dynamics and behavior; (4) Various forms of hierarchical structure; (5) Emergent behavior; (6) Self-organization; (7) Co-evolution with other complex entities or the environment.

The concepts of emergence and self-organization are of particular significance in the scope of this work. *Emergence* in CAS refers to the occurrence of properties and behavior in a system that are not present in the constituent components, i.e., global behaviors are emergent results of local interactions. Similarly, *Self-Organization* is the emergence of global coherence out of local interactions. Natural instances of self-organization include the swarming formation of birds in flight, and the emergence of cognitive abilities from interactions of neurons in the brain. In the context of smart cities, the flow of vehicular traffic is a prominent instance of self-organized behavior which can be guided by smart traffic management systems via traffic signals [13].

B. Vulnerability and Resilience of CAS

The resilience of complex systems has been the subject of active research in diverse disciplines, ranging from ecology [14] to power distribution systems [7] and even smart cities [5]. Yet, the bulk of available literature on this topic emphasize on resilience of CAS to naturally occurring and random perturbations. Amid the spectrum of definitions of resilience proposed in such works [15], one of the most general is given by Hollangel et al. [16] as: "The ability of a system to endure failure and recover from mishaps by restoring its capacities". This definition captures the objectives of system-level studies, yet it fails to satisfy the requirements of security analyses. While recovery from failure may demonstrate the long-term sustainability of system's operations, the security consequences of short-term failures may be catastrophic. For instance, temporary disruption of power grids or traffic flow, however technically recoverable, may incur severe damages to a city's economy. Therefore, there is a crucial need for security-oriented alternatives to this definition.

Similarly, the concept of vulnerability in CAS is defined either too loosely, or too ad-hoc. For instance, [17] defines vulnerability as the system's inability to resist stresses, which may be exploited by threats and hazards. On the other hand, [18] provides a network-oriented definition as links or nodes whose removal adversely impact the functions of a complex network. It is evident that a generic and quantitative definition of vulnerability is needed to form the basis of a computational framework for analysis and measurement of security in CAS.

C. Modeling Approaches to CASC

Abstraction and capturing the dynamics of complex smart cities is an active topic of research, with many proposals and modeling frameworks developed through the past decades. However, the bulk of such approaches may be fitted within three classes of models, namely *dynamical systems, agent-based models, and complex network models* [6]. Having multiple approaches enables various levels of abstraction for high-dimensional CASC, thereby providing multiple perspectives for capturing the structure and dynamics of smart cities in the context of vulnerability analysis.

The first of these approaches is based on the fact that smart cities are dynamical systems, meaning that their states change as a function of time. In this perspective, the CAS dynamics of smart cities can be modeled as [6]:

$$\dot{x}(t) = f(x(t), \beta(t)) \tag{1}$$

Where $\dot{x}(t)$ is the first-order derivative of x with respect to time t, $x = (x_1, x_2, ..., x_n)$ is the n-dimensional state of CAS, β is the state of the environment (or alternatively, control input), and f is the dynamics of the system. The set of all possible configurations of x is termed the *phase space* of the system, henceforth denoted by X. A solution $x(t)$ to the equation 1 constitutes a *trajectory* in phase space. Any trajectory is uniquely defined by the initial conditions, $x(0) \equiv x_0$. Accordingly, the Time-T *Flow* ϕ_T for initial conditions $x(0)$ is defined as $\phi_T(x(0)) = x(T)$.

In dynamical systems, an *attractor* is a bounded region in phase space to which trajectories with certain initial conditions converge or come arbitrarily close. Formally, an attractor is an invariant set $\Lambda \in X$, where trajectories of perturbations that lead to states outside of Λ eventually return to Λ. Attractors may be isolated points, limiting cycles, or more complex objects in the phase space.

A *basin of attraction* $\Omega(\Lambda)$ is the set of all states which fall on trajectories that lead to attractor Λ. Formally,

$$\Omega(\Lambda) = \{x \in X : \lim_{t \to \infty} \phi_t(x) \in \Lambda\} \tag{2}$$

Accordingly, the *basin boundary* $\partial\Omega$ of a CAS is defined as the set of states that are not in any basin. Formally:

$$\partial\Omega = X - \bigcup_i \Omega(\Lambda^i) \tag{3}$$

Even though the dynamical model provides a fundamental mathematical perspective on the behavior of CAS, the abstraction and computational aspects of this model are prone to the curse of dimensionality and thus are severely restricted in high-dimensional smart city systems. Therefore, alternative models are often used to simplify the dynamical representation

and abstraction of such CAS. Accordingly, agent-based models are constructed by capturing the behavior of individual components in the system, and employing tools from game theory (e.g., [10]) and similar fields to roll-out the natural dynamics of interactions among those components. Alternatively, network-based models adopt a connectionist approach to construct a network abstraction based on the relationships and interactions of various components in the system (e.g., [19]).

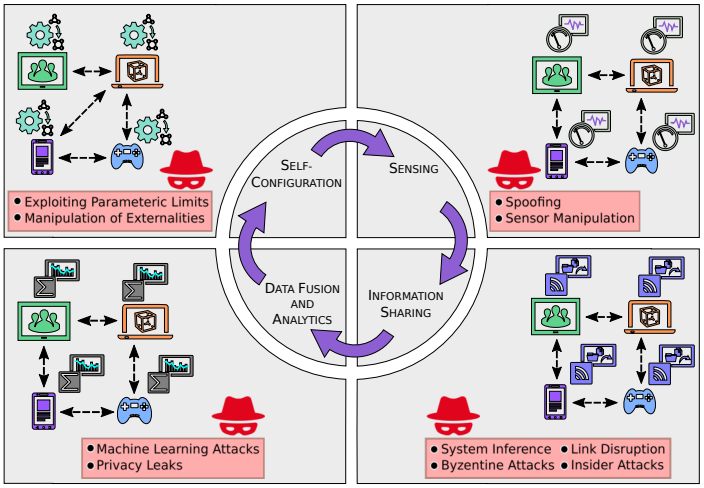

Fig. 1: Instances of potential attacks on smart cities.

III. Threat Model and Metrics

The adaptive dynamics of smart cities gives rise to a variety of vulnerabilities and attack surfaces. By definition, the macro-scale behavior of such systems is the emergent result of micro-scale actions of local or individual elements. Therefore, adversarial perturbations of micro-scale structure and dynamics may result in amplification of perturbations and manipulation of the macro-scale behavior.

To ensure a consistent and comprehensive study of such attacks, we first develop suitable definitions of attack, vulnerability, and resilience in CAS. We differentiate between two types of attacks, namely passive and active attacks. *Passive attacks* aim at exposure of structural and dynamical properties of the targeted CAS, and do not require exertion of additional input to the system. Instances of such attacks are network traffic analysis [20] and inference of dynamics [19]. On the other hand, *Active attacks* involve the implementation of adversarial actions to achieve an adversarial objective. Building on the dynamical model of Section II-C, we define *adversarial action* as the intentional manipulation of either the state or dynamics of a CASC system, such that the resulting state-space trajectory passes through states, which may include states outside of desired basins of attraction, states within undesired submanifold of the phase space (e.g., undesired basins of attraction), or ill-defined states within a modified phase space. Accordingly, the modes of adversarial actions can be categorized as those perturbing the state configuration of CASC, and those that manipulate the dynamics of CASC, formalized as follows:

1) State Manipulation: Let $\gamma(x_t)$ be the perturbation to state $x_t \in X$, i.e., the perturbed state is obtained via $x_t^p = x_t + \gamma(x_t)$. The problem of adversarial state manipulation is to devise the function $\gamma(x_t)$ such that at an arbitrary time T:

$$x_T = \int_{t_0}^{T} f(x_t + \gamma(x_t), \beta_t)dt \in X^* \tag{4}$$

Where t_0 is the initial time, and $X^* \in X'$ is the set of states within the space of undesired states X' which conform to adversarial objectives. It is noteworthy that a sustainable impact is imposed when the adversary aims for driving the target into X^*'s basins of attraction.

Alternatively, if the objective is to reach specific trajectories $\mu(t)$ in the space of undesired trajectories M rather than particular states, the problem can be rearranged as devising $\gamma(x_t)$ s.t. some measure of distance between the original and desired trajectory becomes smaller than an arbitrary error threshold ϵ, i.e.,

$$\|\dot{x}(t) - \dot{\mu}(t)\| = \|F(x_t + \gamma(x_t), \beta_t) - \dot{\mu}(t)\| < \epsilon$$

2) Dynamics Manipulation: Let $\lambda(x_t, \beta_t)$ be the perturbation to the environment (alternatively, it can be viewed as control input). The problem of adversarial dynamics manipulation is to devise a suitable control perturbation $\lambda(x_t, \beta_t)$, such that at an arbitrary time T:

$$x_T = \int_{t_0}^{T} f(x_t, \beta_t + \lambda(x_t, \beta_t))dt \in X^* \tag{5}$$

It must be noted that X^* is not necessarily a subset of X, as the phase space may shift due to perturbations. Alternatively, the problem of reaching specific trajectories can be formulated similarly to the case of state manipulation, with the following optimization objective:

$$\|\dot{x}(t) - \dot{\mu}(t)\| = \|F(x_t, \beta_t + \lambda(x_t)) - \dot{\mu}(t)\| < \epsilon$$

With the concept of attack formalized, we can construct suitable measures of vulnerability and resilience on the same grounds. We adopt a well-established fact from the realm of cyber-security that no system can be completely secure against all possible attacks. Hence, the objective of securing a system becomes deterrence of attacks in an economic sense, namely making successful attacks as costly as possible. Accordingly, we define the *vulnerability* of an element (state, trajectory, or dynamics) in a CASC to a specific adversarial action, as the inverse of the minimum amount of cost incurred to the adversary to impose the maximum achievable cost to the targeted CASC, via implementing the adversarial action on the designated element. This definition assumes that adversarial cost $C_{adv} \geq 1$, and hence the value of vulnerability is in the range $[0, 1]$, whose unit is determined by the dimensions of adversarial cost C_{adv}. In a similar manner, we define the *resilience* of a CAS against a certain attack as the minimum cost imposed on the adversary to successfully implement that adversarial action and force the CAS into an undesired state or trajectory. The selection of adversarial and CASC cost metrics is highly dependent on the context of analysis. One simple instance of choices for adversarial cost can be the minimum number of perturbations required for a successful attack. A similar choice for the CASC cost is the loss of connectivity in the network model of its interactions.

IV. Classification of Attack Surfaces

Attack surfaces are structural and dynamical components of CASC that may be targeted in active and passive attacks. In this section, we present three schemes categorizing such components, and provide attack instances for each identified component.

1) CIA-based: The first approach concentrates on the security dimensions being attacked. The general dimensions of security are Confidentiality, Integrity, and Availability, forming the CIA triad of security. *Confidentiality* refers to the

restriction of unauthorized access to protected information. Examples of attacks on confidentiality in CAS include the inference of states, dynamics, and interaction protocols in a self-organizing swarm of UAVs. *Integrity* is maintaining and assuring the accurate functioning of the system in the intended manner. An instance of corresponding attacks is manipulation of a distributed autonomous navigation system to induce collisions. *Availability* is assuring the uninterrupted operation of the system. Induction of cascade failures in power distribution systems is a well-established instance of such attacks on CASC.

2) DDDAS-based: Another approach to classification of attack surfaces is based on the paradigm of Dynamic Data-Driven Application Systems (DDDAS). The decentralized adaptive behavior of CASC implies the existence of a feedback control loop in the constituent components. Accordingly, each component of such CACS monitors the changes in its environment, analyzes the observations and its internal state with respect to local rules and objectives, and adjusts its operating parameters accordingly. This process can be accurately captured within the framework of DDDAS. A DDDAS is a symbiotic feedback control system, which can dynamically analyze the state of system and its environment to control and determine when, where, and how it is best to gather additional data, and in reverse, can dynamically steer the applications based on the obtained measurements [21]. The operational cycle of an element in a generic distributed smart city system comprises of 4 components:

- *Sensing*: Observing the state of agent's environment and retrieving relevant information that may be disseminated by other agents
- *Information Sharing*: Communicating agent's current state and observations with other agents
- *Data Fusion and Analytics*: Integration and processing of observed and retrieved information
- *Self-Configuration*: Configuration of agent's functional parameters according to processed information

As illustrated in Figure 1, each component of the DDDAS cycle constitutes attack surfaces that can be the subject of adversarial actions targeting one or a combination of the CIA dimensions. However, as shown in Table I, under this schemes some attacks may find overlapping roots between different component.

3) Functionality-based: We also propose a more general functionality-based approach to classification. The building blocks of CAS are its structure and topology, dynamics of interactions, and the internal dynamics of each constituent agent. Accordingly, we further categorize the attack surfaces of CAS into those stemming from the *Network Structure, Cooperation Protocolos,* or *Actuation Functions*, detailed below:

A. Attacking the Network Structure

As discussed in Section II-C, CAS can be modeled as networks of interacting agents. Depending on the model's context and objective, this network may represent the communication links between agents, their interactions, dependencies, or other types of relationships. As is the case with distributed networked systems, such as communications and social networks, the intrinsic network structure of CASC gives rise to a number of potential vulnerabilities that can be exploited to mount passive and active attacks against the system. By means of *traffic analysis* [20] and *inference* attacks [19], adversaries can target the confidentiality dimension of CASC to identify the topology and dynamics of their networks. Knowledge of the network topology enables the adversaries to optimize denial of service attacks by analyzing the structure of their target and determining the most critical regions [20]. To further expand on this surface, consider the case of a self-organizing swarm of UAVs, as illustrated in Figure 2. The inter-UAV network depicted in this figure is a graph with 2 hubs (i.e., Nodes 3 and 4), through which a large portion of network flows pass. If the adversary aims a jamming attack at only these two hubs, the network becomes completely disconnected, thereby the entire operation of the system is disrupted at minimal cost to the adversary. Under certain circumstances, this type of attack may cause cascading effects that result in total system failure over time. As example of which is cascade failures in power grids, further detailed in Section VI.

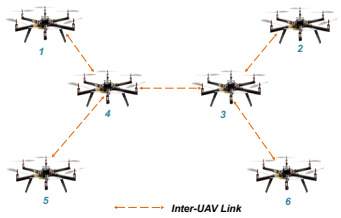

Fig. 2: Example of topological vulnerability

B. Attacking Cooperation Protocols

Considering the independent and self-interested nature of agents in CASC, stabilization and efficiency of many real-world applications of such systems necessitate the implementation of rules and protocols to induce and maintain cooperative interactions between agents. For instance, formation control and navigation of UAV swarms require the sharing of positional information among UAVs, as well as their coordination of navigational parameters. Implementation of cooperation protocol creates another source of attack surfaces. Adversaries may target the confidentiality of CASC via passive sniffing of shared information through either insider and outsider attacks. This type of passive eavesdropping enables further active attacks through inference and identification of objectives and system dynamics.

The integrity of such systems can be targeted in various ways. By spoofing legitimate agents, adversaries can inject false data into the information sharing pipeline of CAS. Also, spoofed, compromised, or malicious insider agents may falsify their resource requirements, or even pose as several agents to gain unfair access to shared resources. In the domain of distributed wireless networks, this type of exploitation is known as *Sybil attack* [22]. Furthermore, in systems with constrained information sharing capacities, adversarial perturbation of the environment may lead to sharing of incorrect or incomplete information. For instance, consider the case of a UAV swarm which relies on individual reporting of observed obstacles for collision avoidance. If the reporting protocol limits the number of reported obstacles to the n nearest objects observed by a UAV, an adversary may spoof or generate $m >> n$ minor obstacles in the vicinity of the UAV to prevent it from informing rest of the swarm about major nearby obstacles.

Attacks on the availability aspect may also come in different forms. Spoofed, compromised, or malicious insider agents may act as information *blackholes* [22] by tactically refusing to share their information at particular times. In CASC that

rely on multi-hop communications, this attack can be more damaging if the agent stops forwarding information received from other neighbors as well. Another type of attack is based on spoofed, compromised, or malicious insider agents disseminating certain information that cause termination of cooperation. In our example of UAV swarm, transmission of messages such as "mission accomplished", "mission failed", or radio silence signal in tactical scenarios, may cause the cooperative process to end. Furthermore, if the cooperation protocol is not well-designed, broadcast of certain resource constraints or environmental conditions may result in prevalence of agents' selfishness over cooperation. This condition may be induced through either dissemination of fake information, or adversarial manipulation of the environment [19].

C. Attacks on Actuation Functions

The main objectives of CASC are realized by each agent via actuation functions. In the example of UAVs, actuation functions are cyber-physical controllers of motion and communications. In general, the ultimate goal of all attacks introduced so far is indirect manipulation or disruption of actuation functions. Adversaries may also directly target the actuation of CASC through attack surfaces in actuation mechanisms and functions. Mounting attacks on confidentiality of actuation may be in the form of parameter inference. Obtaining knowledge of operating parameters through side-channel attacks enables the adversary to derive a more accurate estimation of system's state and dynamics, thereby allowing the optimization of active attacks against the system. Also, in competitive CAS, complete knowledge of an agent's operating parameters may provide other agents with an unfair advantage. For instance, consider a CAS setup to automate the sharing of information on cyber attacks among corporations [23]. In this scenario, agents aim to share the minimal amount of data required to preserve the long-term benefits of information sharing. If an adversarial agent is able to estimate the parameters used by another agent in filtering and disseminating information, it may allow the adversary to infer the undisclosed portion of agent's information. A sophisticated attack in such incomplete information systems can be the adversarial disclosure of parameters to competitors, thereby causing the system dynamics to diverge from a beneficial equilibrium. Economic and political parallels of this phenomenon are instances of insider trading and whistleblowing (e.g., [24]).

The integrity of actuation functions may be targeted via manipulation of the environment or sensory observations. In an autonomous fleet of self-organizing vehicles, calculated manipulation of the visual input to a vehicle may result in an *adversarial example* [25] for the machine learning component of the system. Adversarial examples are minimally perturbed inputs that cause misclassifications in machine learning algorithms. For instance, minor changes in a speed sign on the side of a street can result in its misclassification as a stop sign by an autonomous vehicle, causing it to stop in an unsafe location [26]. In some cases, even spoofed perturbations of the environment is sufficient for manipulation of actuation functions. A real-world example of such cases is the Automatic Collision Avoidance System (ACAS) utilized by many of today's commercial aircraft [26]. This system generates motion advisories according to the position and heading of other aircraft in the environment, obtained from an unencrypted, open protocol known as ADS-B. An adversary may simply fake the presence and trajectory of nonexistent aircraft by spoofing, ADS-B signals, which can lead to ACAS advisories that change the trajectory of targeted aircraft [26].

Similar attacks can also target the availability of actuation functions. Adversaries may manipulate the environment such that the actuation functions of CASC agents fall within undefined or terminal states. In our UAV example, induction of emergency conditions through environmental or sensory manipulation can drive targeted agents into safe modes, which in many cases trigger automatic Return-to-Base (RTB) or emergency landing procedures [26].

Table I presents the classifications of the sample attacks discussed in this section.

V. Simulation Framework

As an approach towards analysis of impact in attacking the emergent dynamics of CASC, we propose a framework for simulation of adversarial actions against generic CASCs. With the aim of analyzing the maximum impact of attacks, this framework is designed to automatically derive the optimal sequence of adversarial actions against CASC models. Also, our framework supports the analysis of both whitebox and blackbox attacks, meaning that the adversary can be considered to have complete, partial, or no a priori knowledge of the system dynamics. Furthermore, this framework allows for arbitrary designation of adversarial goals (e.g., network disruption, actuation manipulation, etc.), and can be configured for arbitrary types of adversarial actions.

The initial step in this framework is to obtain an estimation of dynamics in the targeted CASC from time-series observations of the system. For simulation of blackbox attacks, this can be achieved through a variety of methods developed for identification of nonlinear dynamics, such as utilization of deep neural network (e.g., [27]) and generative adversarial networks [28]. When partial knowledge of the system is assumed, the estimation technique can be based on a generic model of the dynamics with unknown model parameters, which may be estimated via statistical and machine learning techniques (e.g., [19]). As for the simulation of whitebox attacks, this estimation can be fixed to a complete dynamical model of the system. Examples of each case are presented in Section VI.

With the initial estimate of dynamics at hand, the next step of this framework is to create a secondary simulation of the targeted system in order to obtain the optimal attack policy $\pi^*(S)$, which maps any observed state S of the estimated system to an optimal action A_S. This action corresponds to one the adversarial actions defined in the initial configuration of simulations, Instances of which are node removals for attacks on network structure, sensory overload for attacks on cooperation protocols, and crafting adversarial examples for manipulation of actuation functions.

Accordingly, we propose *reinforcement learning* (RL) as a promising approach to the problem of policy optimization. RL enables the learning of optimal decision making in choosing control (i.e., action) sequences that maximize a certain objective based on some reward signal [29]. A major advantage of RL is in its ability to learn a model of its environment through exploration and trial and error. Further, recent advances in deep RL have demonstrated the feasibility of applying RL to high-dimensional complex environments, such as autonomous driving and playing Atari games [30]. Hence, this approach is a promising candidate for deriving control policies in complex systems.

Functional Surface	Attack Example	CIA Dimension	DDDAS Surface	Attack Type	Attack Mode
Network Structure	Traffic Analysis, Topology Inference	C	IS	Passive	N/A
	Topological Disruption	A	IS	Active	State
	Cascade Induction	I, A	IS, SC	Active	Dynamics
Cooperation Protocols	Sniffing	C	IS	Passive	N/A
	Sybil	I, A	IS, SC, S	Active	State/Dynamics
	Information Manipulation	I, A	SC, AN	Active	Dynamics
Actuation Functions	Parameter/Dynamics Inference	C	IS, SC	Passive	N/A
	Competitive Intelligence	C	IS, SC, AN	Passive	N/A
	Adversarial Examples	I, A	S, AN, SC	Active	State
	Spoofing	I, A	S, AN, SC	Active	State / Dynamics
	Induction of Terminal States	I, A	S, AN, SC	Active	State

TABLE I: Classification of sample attacks - C, I, and A stand for Confidentiality, Integrity and Availability, respectively. For DDDAS attack surfaces, S is Sensing, IS is Information Sharing, AN stands for Analytics, and SC is Self-Configuration.

The RL problem is described by the Markov Decision Process (MDP) tuple (S, A, P, R), where S is the set of reachable states in the process, A is the set of available actions, R is the mapping of transitions to the immediate reward, and P represents the transition probabilities (i.e., system dynamics). At any given time-step t, the MDP is at a state $s_t \in S$, which represents the current configuration of simulated CASC. The reinforcement learning agent's choice of action at time t, $a_t \in A$ causes a transition from s_t to a state s_{t+1} according to the transition probability $P^{a_t}_{s_t, s_{t+a}}$. The agent receives a reward $r_t = R(s_t, a_t) \in \mathbb{R}$ for choosing the action a_t at state s_t.

Interactions of the agent with MDP are captured in a policy $\pi : S \rightarrow A$. The objective of reinforcement learning is to find the optimal policy π^* that maximizes the cumulative reward at any time t, denoted by the return function $\hat{R} = \sum_T^{t'=t} \psi^{t'-t} r_{t'}$, where $\psi < 1$ is the discount factor that accounts for the diminishing worth of rewards obtained further in time, hence ensuring that $\hat{R}$ is bounded.

An approach to this problem is the *Action-Value Function* optimization algorithm or Q-Learning. In every iteration of this technique, the optimal value of each action is calculated as the expected sum of future rewards, assuming that every action taken after the current choice follows the optimal policy. Once the optimal policy is obtained from the secondary simulation, it is implemented on the primary simulation to observe the impact for a user-defined number of time-steps. At this point, the new observations are fed back to the estimation algorithm to improve adversary's model of target dynamics, and derive the optimal attack policy for the updated model. This iterative process is executed until the user-defined criteria for attack success or termination are reached. At every iteration of Q-Learning, the process selects its estimation of the best possible action, which is one of the designated adversarial actions designated in the configuration of attack simulation.

This process is formalized in Algorithm 1. Before execution, this algorithm must be integrated with a dynamical simulation or physical prototype of the target system. Also, the user shall define a technique for estimation of dynamics, designate an attack objective, the set of permissible adversarial actions, the cost function of attack, and the criteria for termination of Q-learning. Upon execution, the algorithm iteratively observes the state of the target system, and updates its estimate of target's dynamics according to a pre-defined technique (line 5). This estimate is then used to create a simulation of target from an adversary's perspective, which is then explored via Q-learning to obtain an optimal attack policy based on current estimate (line 6). This policy is then applied to the original simulation or prototype of the target (line 7), and the simulated adversary's observation of target's state is updated according to the resulting state of the target (line 8). This process is repeated until the adversarial reward reaches the designated attack objective (line 4).

Algorithm 1: Attack Simulation Framework

Input : dynamical simulation, Attack cost function C, objective O, set of actions A, termination criteria X

Data: initial target configuration G_0, reward/cost of attack R, current configuration G, policy π

Output: optimal reward/cost of attack R, final configuration G^*, optimal policy $\pi^*(.)$

1 $R \leftarrow 0$
2 $G \leftarrow G_0$
3 Initialize π to a random distribution
4 **while** $R < O$ **do**
5 $\quad U \leftarrow$ `EstimateDynamics(`G, X`)`
6 $\quad R, \pi \leftarrow$ `QLearning(SimulateDynamics(`G, U, π`),` G, U, X, C`)`
7 $\quad$ Implement $a \leftarrow \pi(G)$
8 $\quad$ Update G
9 **end**

It is noteworthy that this framework can only succeed if the attack objective is reachable from the initial state of the target, and with the defined set of actions. Otherwise, this algorithm will provide a best-effort performance in coming as close as possible to the objective. Also, the accuracy and convergence of this algorithm is heavily dependent on the dynamic estimation mechanism. The choice of estimation technique and its updating criteria must be such that the estimation errors do not consistently accumulate, and remain bounded over a large number of iterations.

Furthermore, Algorithm 1 does not intrinsically account for constraints on execution time, therefore such limitations must be implemented within attach the cost function. Similar to the reachability criteria of optimality, if time constraints of the problem fall below the time required for reaching the optimal answer, this algorithm still performs a best-effort search for optimal attacks and potential impact. Such best-effort results are indeed representative of practical worst-case impact levels under the conditions modeled by user-defined parameters.

VI. Case Studies

To study the performance and feasibility of our proposed framework, we investigated its application to 2 real-world CAS scenarios, namely: Inducing cascade failures in power distribution networks and disruption of traffic flow via localized attacks. For each case study, we describe the objective and classify the type of attack according to the schemes introduced in Section IV. We then report the approach and experimental setup, and present the results in terms of quantitative impact and vulnerability.

A. Cascade Failures in Power Grids

Power distribution networks constitute a well-known instance of CAS [7] that are susceptible to cascading failures triggered by malfunctions in one or more local components, such as relays and transmission lines. In such cases, the load of a failed component is balanced onto neighboring nodes, causing them to overload and fail as well [31]. In this case study, the attack objective is to analyze the maximum possible disconnection of a power network by induction of cascading failures through sequential removal of transmission lines in a simulated power grid. The case of sequential attacks on power grids is recently studied by Yan et al. [31], who also use a an approach based reinforcement learning to analyze the impact of such attacks. One major difference between the methodology of [31] and this case study is the assumption of a blackbox attack in our approach, which circumvents the issues caused by modeling challenges in the study of cascading power grid failures. Moreover, this case study demonstrates an instance of applying a dynamical system model to analysis of vulnerabilities in CASC.

1) Objective and Classification: The objective of this attack is to disconnect the minimum number of transmission lines one at a time, such that the system collapses. This attack targets the network structure to compromise the Availability dimension of CIA by implementing an adversarial action to manipulate the state of this CAS.

2) Experiment Setup: The benchmark network used in this experiment is a mid-size IEEE RTS-79 architecture. This system is comprised of 24 buses, 38 transmission lines, 17 load buses, and 10 generating units, with a total generation capacity of 3405MW, and a peak load of 2850MWs. A line is considered to be alive if it operates with a load that is smaller than its capacity. Once this threshold is reached, the line fails and all of its load is distributed equally among the nodes that are directly connected to it.

The dynamical simulation was implemented in Python using the PyPSA toolbox. Following the setup of [31], the attack objective was set to cause at least 8 lines failures, while minimizing direct disconnection of lines by the attacker, and maximizing the disconnections resulting from cascading failures. We constrained the maximum number of iterations of each simulation to 500, and repeated each full simulation 100 times. As for the estimation method, we adopted the architecture proposed in [32] for a convex-based Long-Short Term Memory (LSTM) neural network to approximate the nonlinear dynamics of the power grid.

Figure 3 depicts the obtained results, avereged over 100 repetitions. It can be seen that our framework achieves an outage of 8.6 lines with only 3 direct node removals, thereby demonstrating the applicability of our framework in simulating emergent attacks in real-world CAS. Accordingly, the vulnerability measure of this network structure to node removal attacks is $\frac{1}{3} = 0.34$.

B. Disruption of Traffic Flow

Recent studies have shown that traffic signals are vulnerable to a variety of attacks, ranging from physical tampering to remote bypassing of authentication and triggering of fault-handling mechanisms [33]. Such vulnerabilities provide motivated adversaries with the means for disruption of the traffic flow in a city by sequential tampering of a few signals [13]. In many traffic signals, fail-safe mechanisms prevent an adversary from directly manipulating the signal. However, it is still possible for the attacker to tamper with the scheduling of compromised traffic signals. This can then be used to maximize the impact of attack in terms of minimizing city-wide traffic flow. Laszka et al. [13] report a study on the vulnerability of smart cities to such attacks and propose a heuristic algorithm to analyze the impact of sequential tampering of traffic signals in maximizing traffic flow. To demonstrate the generality of our framework, we perform a similar experiment with the same premises as that of [13] and compare the efficacy of their algorithm with our RL-based proposal.

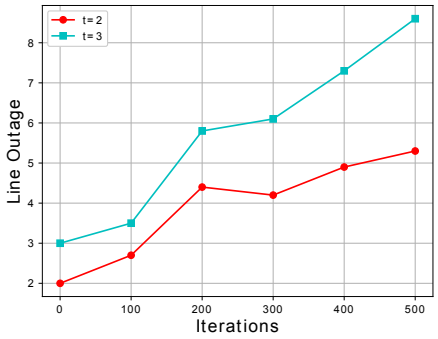

Fig. 3: Induction of cascade failure in power grids via direct targeting of t=2 and t=3 lines

1) Objective and Classification: The objective of this attack is to tamper with the internal schedules of traffic signals such that the total travel time in the transportation network is dramatically increased. Similar to [13], we assume that the attacker can compromise at most $B \leq |S|$ intersections at any given time, where $|S|$ is the total number of intersections. This attack targets the network flow and structure to compromise the Availability dimension of CIA by implementing an adversarial action to manipulate the state of this CASC.

2) Experiment Setup: To maintain the ability of comparative analysis, our experiment setup follows that of [13]: we performed traffic flow simulations in SUMO (Simulation of Urban MObility)[1] using the same map as that of [13] with 5 major intersections as possible targets S for the attack. The default configurations for these traffic signals were selected based on [13]'s parameters which aim to minimize total travel time during normal operation [2].

Considering a model of traffic flow that represents peak commutes during an afternoon, we measure the average travel time as the metric of effectiveness for the attack. Figure 4 depicts the obtained results, avereged over 100 repetitions. It illustrates the performance of our proposed framework in analyzing such attacks compared to that of [13]. It can be seen that our generic approach performs as well as that of a heuristic algorithm developed for the particular case of traffic flow attacks, thereby verifying the general applicability of our framework.

VII. Conclusion

We introduced the paradigm of adversarial attacks targeting the nature of dynamics in Complex Adaptive Smart Cities (CASCs). Aiming to develop a comprehensive foundation for analysis of such attacks, we presented three approaches to the modeling of CASC as dynamical, data-driven, and game-theoretic systems. We developed suitable definitions of attack,

[1] http://sumo.dlr.de/

[2] http://aronlaszka.com/data/laszka2016vulnerability.zip

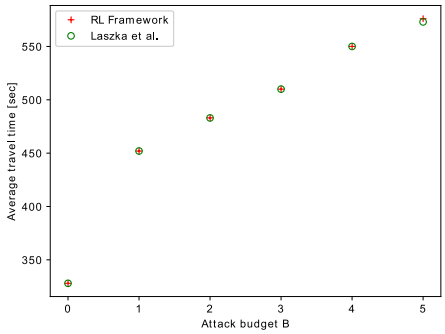

Fig. 4: Maximal traffic disruption via resource-constrained tampering of traffic signals

vulnerability, and resilience in the context of CASC Security, and introduced three schemes for classifying threats based on security dimensions, data-driven abstraction, and fundamental functionalities of CASC. Building on this foundation, we proposed a framework for simulation and analysis of attacks on CASC, and demonstrated its performance in vulnerability analysis of power grids and transportation networks. These case studies also demonstrate the need for novel techniques and methodologies for threat detection and mitigation in CASC.

Acknowledgements

This work was supported by the National Science Foundation (NSF) (NSF-CRII-CPS-1743490). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

References

[1] R. Khatoun and S. Zeadally, "Smart cities: concepts, architectures, research opportunities," *Communications of the ACM*, vol. 59, no. 8, pp. 46–57, 2016.

[2] R. Petrolo, V. Loscri, and N. Mitton, "Towards a smart city based on cloud of things," in *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities*, pp. 61–66, ACM, 2014.

[3] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, and P. Siano, "Iot-based smart cities: a survey," in *Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on*, pp. 1–6, IEEE, 2016.

[4] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[5] K. C. Desouza and T. H. Flanery, "Designing, planning, and managing resilient cities: A conceptual framework," *Cities*, vol. 35, pp. 89–99, 2013.

[6] M. Mitchell, *Complexity: A guided tour*. Oxford University Press, 2009.

[7] G. A. Pagani and M. Aiello, "The power grid as a complex network: a survey," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 11, pp. 2688–2700, 2013.

[8] E. Ordoukhanian and A. M. Madni, "Resilient multi-uav operations: Key concepts and challenges," in *54th AIAA Aerospace Sciences Meeting*, p. 475, 2016.

[9] S. Mittal and J. L. Risco-Martín, "Simulation-based complex adaptive systems," in *Guide to Simulation-Based Disciplines*, pp. 127–150, Springer, 2017.

[10] V. Behzadan and B. Rekabdar, "A game-theoretic model for analysis and design of self-organization mechanisms in iot," *arXiv preprint arXiv:1701.04562*, 2017.

[11] J. H. Miller and S. E. Page, *Complex adaptive systems: an introduction to computational models of social life: an introduction to computational models of social life*. Princeton university press, 2009.

[12] S. Lloyd, "Measures of complexity: a nonexhaustive list," *IEEE Control Systems Magazine*, vol. 21, no. 4, pp. 7–8, 2001.

[13] A. Laszka, B. Potteiger, Y. Vorobeychik, S. Amin, and X. Koutsoukos, "Vulnerability of transportation networks to traffic-signal tampering," in *Proceedings of the 7th International Conference on Cyber-Physical Systems*, p. 16, IEEE Press, 2016.

[14] B. Walker, C. S. Holling, S. Carpenter, and A. Kinzig, "Resilience, adaptability and transformability in social–ecological systems," *Ecology and society*, vol. 9, no. 2, 2004.

[15] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and measures of system resilience," *Reliability Engineering & System Safety*, vol. 145, pp. 47–61, 2016.

[16] E. Hollnagel, D. D. Woods, and N. Leveson, *Resilience engineering: Concepts and precepts*. Ashgate Publishing, Ltd., 2007.

[17] N. Pedroni, *Advanced Methods for the Risk, Vulnerability and Resilience Assessment of Safety-Critical Engineering Components, Systems and Infrastructures, in the Presence of Uncertainties*. PhD thesis, Grenoble 1 UGA-Université Grenoble Alpes, 2016.

[18] C. Moore, J. Grewar, and G. S. Cumming, "Quantifying network resilience: comparison before and after a major perturbation shows strengths and limitations of network metrics," *Journal of Applied Ecology*, vol. 53, no. 3, pp. 636–645, 2016.

[19] V. Behzadan, A. Noormohammadi, M. Yuksel, and M. Gunes, "A novel framework for strategic destabilization of dynamic terrorist networks," *Advances in Social Networks Analysis and Mining (ASONAM), IEEE/ACM International Conference on.*, 2017.

[20] V. Behzadan, "Real-time inference of topological structure and vulnerabilities for adaptive jamming against covert ad hoc networks," Master's thesis, University of Nevada, Reno, 2016.

[21] R. M. Fujimoto, N. Celik, H. Damgacioglu, M. Hunter, D. Jin, Y.-J. Son, and J. Xu, "Dynamic data driven application systems for smart cities and urban infrastructures," in *Winter Simulation Conference (WSC), 2016*, pp. 1143–1157, IEEE, 2016.

[22] A.-S. K. Pathan, *Security of self-organizing networks: MANET, WSN, WMN, VANET*. CRC press, 2016.

[23] I. Vakilinia, S. Sengupta, *et al.*, "Evolving sharing strategies in cybersecurity information exchange framework," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 309–310, ACM, 2017.

[24] L. A. Smales and M. Thul, "A game theory model of regulatory response to insider trading," *Applied Economics Letters*, vol. 24, no. 7, pp. 448–455, 2017.

[25] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint arXiv:1602.02697*, 2016.

[26] V. Behzadan, "Cyber-physical attacks on uas networks-challenges and open research problems," *arXiv preprint arXiv:1702.01251*, 2017.

[27] O. Ogunmolu, X. Gu, S. Jiang, and N. Gans, "Nonlinear systems identification using deep dynamic neural networks," *arXiv preprint arXiv:1610.01439*, 2016.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[29] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[30] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint arXiv:1708.05866*, 2017.

[31] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 200–210, 2017.

[32] Y. Wang, "A new concept using lstm neural networks for dynamic system identification," in *American Control Conference (ACC), 2017*, pp. 5324–5329, IEEE, 2017.

[33] J. Reilly, S. Martin, M. Payer, *et al.*, "On cybersecurity of freeway control systems: Analysis of coordinated ramp metering attacks 2," in *Transportation Research Board 94th Annual Meeting*, 1755.