The Cybersecurity Dynamics Way of Thinking and Landscape (Invited Paper)

Shouhuai Xu
Laboratory for Cybersecurity Dynamics
Department of Computer Science, University of Texas at San Antonio
shouhuai.xu@utsa.edu

ABSTRACT

The Cybersecurity Dynamics framework offers an approach to systematically understanding, characterizing, quantifying and managing cybersecurity from a holistic perspective. The framework looks into cyberspace through the *dynamics* lens because environments in cyberspace often evolve with time (e.g., software vulnerabilities, attack capabilities, defense capabilities, and cybersecurity states). The dynamics lens offers a unique viewpoint, which guides the modeling of the various situations which evolve with respect to cybersecurity. This type of evolution is driven by attackers, defenders, and users of related systems and is manifested by their attack/defense/use activities. Since its inception in 2014, there has been significant progress in characterizing and taming various kinds of cybersecurity dynamics. In this paper we discuss the landscape and way-of-thinking that guide the Cybersecurity Dynamics model, including two killer applications and the technical barriers that serve as outstanding open problems for future research.

CCS CONCEPTS

• **Security and privacy → Formal security models**.

KEYWORDS

Cybersecurity Foundation; Science of Cybersecurity; Cybersecurity Dynamics; Cybersecurity Metrics; Cybersecurity Quantification; Cybersecurity Risk Management; Cyber Defense Decision-Making; Cyber Defense Orchestration; Cyber Defense OODA Loop

ACM Reference Format:
Shouhuai Xu. 2020. The Cybersecurity Dynamics Way of Thinking and Landscape (Invited Paper). In *7th ACM Workshop on Moving Target Defense (MTD'20), November 9, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411496.3421225

1 INTRODUCTION

Computer, information, and network security research has been driven by concepts like confidentiality, integrity, and availability for decades. These concepts are important, as manifested by, for example, the beautiful body of knowledge in modern cryptography, especially the notion of *provable security* using cryptographic primitives and protocols. However, these concepts are limited in the sense that they disregard the *time* dimension, which is critical for real-world cyber defenders. The time dimension is important because many things in cyberspace naturally evolve with time, including cyber threats, cyber defense postures and cybersecurity states. Moreover, the sooner an attack is detected or blocked, the less damage it is able to inflict. This suggests that properly treating the *time* dimension in cybersecurity is an important issue, which served as one motivation for the Cybersecurity Dynamics approach.

The Cybersecurity Dynamics approach is also motivated by a long-standing open problem: How can we *quantify* cybersecurity from a holistic perspective? The importance of quantifying cybersecurity can never be overestimated because solving this problem would automatically answer many risk-management and decision-making questions encountered by practitioners. For example, in order to make conscious investment decisions, a chief executive officer would need to know the *estimated return* of investing a certain amount of money to enhance the enterprise's cyber defense posture (e.g., hiring more defenders or purchasing more defense tools). The *estimated return* does not have to, and perhaps should not, be a singular static value because of the various kinds of *uncertainty* that are involved; instead, it can be an estimated statistical distribution of potential return in risk mitigation or damage control. Similarly, a cyber defense decision-maker would need to choose the best action in order to achieve the desired mission assurance. This is a challenging task because these actions may cause different kinds of *cascading* effects on the network.

The Cybersecurity Dynamics approach is further motivated by the observation that there is no systematic body of cybersecurity knowledge in general; in other words, the cybersecurity knowledge has been fragmented into many sub-disciplines. For example, it would be important to know what is the network-wide or even cyberspace-wide security effect and consequence of a newly invented technique in the various subdomains of programming languages, software engineering or computer architecture. This highlights the importance of a unified framework for structuring and systematizing the body of cybersecurity knowledge. As an analogy, this unified body of knowledge should achieve a systematic and holistic structure akin to what has been achieved in medical science (at least to a large extent): When a new medicine is invented, it comes with a specification on when the medicine is useful and what the side-effects may be.

The Cybersecurity Dynamics approach is first explicitly articulated in 2014 [110, 111], but the endeavor started more than a decade earlier. Corresponding to the motivations mentioned above, this approach is centered at quantifying cybersecurity from a holistic perspective while explicitly accommodating the time dimension

MTD'20, November 9, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8085-0/20/11…$15.00
https://doi.org/10.1145/3411496.3421225

(e.g., the resulting quantification of cybersecurity attributes evolves with time), while aiming to provide an overarching framework for structuring and systematizing the cybersecurity knowledge (i.e., *connecting the dots*). The approach treats evolutions in cyberspace as "natural" phenomena, which emerge from interactions with attackers, defenders, and users as manifested by their attack/defense/use activities in cyberspace.

1.1 Our Contributions

In this paper, we systematically describe the Cybersecurity Dynamics way of thinking, which represents a paradigm shift from the *time-independent* treatment of cybersecurity to the *time-dependent* treatment of cybersecurity. In particular, we introduce a *dynamics layer* for quantifying cybersecurity and supporting various kinds of cybersecurity applications (e.g., cybersecurity risk management and cyber defense decision-making). Then, we describe the Cybersecurity Dynamics landscape corresponding to the Cybersecurity Dynamics way of thinking. We outline ranges of outstanding open problems for future research, indicating that the current understanding of Cybersecurity Dynamics is just the tip of a huge iceberg. We hope that the present paper will motivate or inspire many researchers to tackle these open problems.

1.2 Related Work

The Cybersecurity Dynamics approach is first articulated in [110] and later refined in [112]. The idea of looking at cybersecurity through the lens of emergent properties/behaviors is discussed in [111]. Taking a different perspective than [112], this paper describes the Cybersecurity Dynamics way of thinking and landscape. To the best of our knowledge, the Cybersecurity Dynamics approach offers the first framework aiming at building a foundation for systematically understanding and quantifying cybersecurity, the importance of which is underscored by the significant attention it has received in the literature [38, 50, 87, 88, 91].

As discussed in [111, 112], the first-principle or theoretical cybersecurity research under the umbrella of Cybersecurity Dynamics has been influenced by: (i) biological epidemiology [1, 2, 40, 48, 67], which is first adapted to cyberspace by Kephart and White [46, 47] and further extended to accommodate heterogeneous network structures via degree distributions [3, 71, 73, 80, 80, 81] or adjacent matrices [29, 96, 100]; (ii) interacting particle systems [62], which study the collective behaviors and phenomena that emerge from interacting components; (iii) microfoundation in economics [42], which aims to bridge the gaps between macroeconomics and microeconomics. However, the Cybersecurity Dynamics approach goes far beyond these aspiring disciplines because cybersecurity encounters many issues that have no counterparts in those contexts, such as the unique technical barriers that will be discussed in Section 3.4.3. This can also be manifested by the dynamics layer of abstraction that is presented in the present paper.

1.3 Paper Outline

The rest of the paper is organized as follows. Section 2 describes the Cybersecurity Dynamics way of thinking. Section 3 unfolds the Cybersecurity Dynamics landscape. Section 4 concludes the present paper.

2 THE CYBERSECURITY DYNAMICS WAY OF THINKING

In this paper, we use the term *networks* broadly to accommodate real-world cyber-enabled networks, including enterprise Information Technology (IT) infrastructures, cyber-physical systems, cloud end-to-end computing systems, and military tactical networks (e.g., mobile ad hoc networks or MANETs). Two common features of these networks are: (i) their functions can be affected by cyber attacks, which may be waged remotely from one part to another part of a network; and (ii) there are always some humans in the loop, including attackers, defenders and users. Therefore, we will use the term *networks* and *cyber-physical-human systems* interchangeably. Similarly, we use the terms *cybersecurity* and *security* interchangeably, which are used in a broader sense than the traditional notions of confidentiality, integrity and availability.

2.1 Inspirations for Seeking a New Way of Thinking at a Higher Level of Abstraction

The success of a discipline, especially the creation of a systematic body of knowledge, is often driven by the pursuit of some fundamental concept. For example, the discipline of Cryptography, which itself is a sub-field of cybersecurity, can be seen as built on top of, or centered at, the concept of *indistinguishability*. In the very first context of cryptography, namely encryption, this means that if an attacker cannot distinguish ciphertexts from random bitstrings, an encryption scheme "hides" the plaintext messages well because the attacker cannot learn *anything* useful about the plaintext messages from the ciphertexts, where *anything* can be formulated in the Information Theory framework [90] and the Computational Complexity Theory framework [31]. The explosive development of modern Cryptography in the past decades is ostensibly centered at this concept (e.g., the beautiful notion of *zero-knowledge* [32] and the numerous ideas that are built on top of it). This highlights the importance of seeking fundamental concept(s) to deepen our understanding of cybersecurity. Moreover, there are already some fundamental concepts for computer, information, and network security, especially *confidentiality*, *integrity*, and *availability*, which have driven the security research and practice in the past decades. However, these concepts have their limitations: availability often contextualizes data, systems or services, but confidentiality and integrity are often only concerned with data.

The existing body of cybersecurity knowledge and the accompanying techniques, such as those mentioned above, are mostly geared towards building-blocks, which are absolutely important and actually pave the way for any further developments. We argue [110–112] that for cybersecurity, understanding its systems-oriented or holistic properties is as important as understanding the underlying building-blocks. Holistic cybersecurity considers a much broader context and could enrich the investigation of building-blocks. For example, cyber defenders should be able to know the current situation (i.e., situational awareness) and correspondingly adjust their defense postures. As a more concrete example, suppose the integrity of a data item is compromised at time t_1 and the compromise is detected at time t_2 where $t_1 < t_2$; then, the defender needs to recover the compromised data item, specifically from the time before

the compromise at t_1. Since the integrity of the data item is compromised during the time interval $[t_1, t_2]$, its damage needs to be assessed from a system-wide standpoint. The potential damage may be large enough to warrant an investment to investigate better techniques to effectively mitigate the damage. This hints the importance of considering the time dimension explicitly.

2.2 The Cybersecurity Dynamics Way of Thinking

The preceding discussion suggests the importance of explicitly considering the time dimension, or the shift from a *time-independent* paradigm to a *time-dependent* paradigm. Figure 1 highlights the Cybersecurity Dynamics way of thinking in multiple levels of abstractions, which are collectively called the *dynamics layer*. Below the dynamics layer is the real-world cyber-physical-human systems or networks, which include (as mentioned above) traditional enterprise IT infrastructure, cloud-end-end computing systems, and tactical networks as well as their security architectures and mechanisms. Above the dynamics layer is the application layer, which includes various kinds of applications such as quantitative decision-making, quantitative Command-and-Control (C2), and quantitative risk management. These levels of abstractions are elaborated below and the applications will be elaborated later.

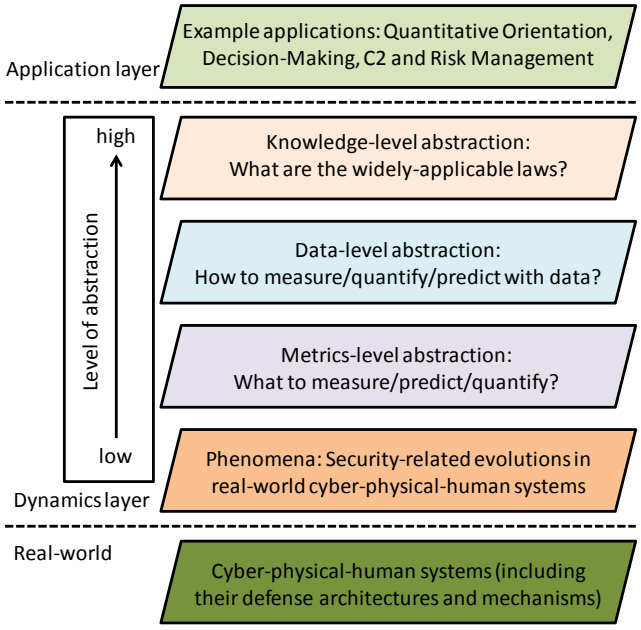

Figure 1: The Cybersecurity Dynamics way of thinking, where the knowledge-level abstractions and the data-level abstraction interact with each other.

2.2.1 What Are the Phenomena to Model? The dynamics layer has multiple levels of abstractions. The lowest level of abstraction corresponds to the identification of dynamics *phenomena*, namely security-related evolutions in networks (or cyber-physical-human systems). Evolutions can be defined with respect to cybersecurity attributes of interest. For example, corresponding to adaptations in employing defense mechanisms over the time horizon, we can define *defense dynamics*; corresponding to adaptations in employing attack mechanisms over the time horizon, we can define *attack dynamics*; corresponding to the evolution of defenders, we can define *defender dynamics* because defenders may have different defense skills and may have different human factors; corresponding to the evolution of attackers, we can similarly define *attacker dynamics*; corresponding to the evolution of users (e.g., employees joining and leaving), we can define *user dynamics*; corresponding to the evolution of the network-wide (or global) cybersecurity state, we can define various kinds of *attack-defense dynamics*; corresponding to the human-system interactions, we can define *human-system* dynamics. These examples highlight that security-related dynamics are a universal phenomenon, while noting that some dynamics can be directly observed (e.g., defender dynamics) but others may not (e.g., attacker dynamics).

2.2.2 What Is the Metrics-level Abstraction? The metrics-level abstraction deals with *defining* the metrics that should be measured. In principle, these metrics are needed for describing the evolution phenomena, for understanding the dynamics, and for serving the needs of the higher-level abstractions. Using the weather as an analogy, these metrics would correspond to those that measure physical phenomena (e.g., temperature), those that model dynamic processes (e.g., a hurricane's eye), and those that attempt to infer the laws that govern such processes (e.g., air pressure dynamics).

From the measurement perspective, some metrics may be measured directly based on observations of the dynamics. For example, let us consider the following simple but important metric: *which or how many computers in a network of n computers are compromised at a* particular *point in time t?* In principle, this metric can be measured by the defender. However, this metric may not be feasible to measure in practice because the measurement can incur false-positives and/or false-negatives, and because n can be very large (i.e., infeasible to have enough experts to thoroughly examine the security states of these n computers, even if the experts are error-free). Nevertheless, this metric is important because it (i) reflects *knowing yourself* in cybersecurity situational awareness, (ii) tells the defender whether the employed defense mechanism is effective or not, and (iii) offers a foundation for evaluating the effectiveness of higher-level defense mechanisms. One such mechanism is a Byzantine fault-tolerant application or service, which can only assure security when the number of compromised replicas are below a certain threshold, meaning that when this threshold assumption is violated, the security assurance is lost. Contributing to the difficulty of this problem is the need to measure this metric for *any* time t, rather than for a *particular* t. This need is both natural and important because the cybersecurity situation, as reflected by this metric, is evolving with time. Moreover, there are many metrics that may not be directly measurable, such as the metrics that describe *attacker capabilities* because the defender may not know exactly what exploits are available to the attacker. These matters highlight the inherent uncertainty in the metrics measurements.

2.2.3 What Is the Data-level Abstraction? The data-level abstraction deals with the various data that need to be collected from the cyber-physical-human systems in the real world to measure the

cybersecurity metrics defined at the metrics-level of abstraction. For example, in order to measure the *attack capability* metric, the defender would need to know the number and kind of attacks that are waged against the network at *any* point in time t. This would allow the defender to build models to predict or forecast the attacks that may be waged against the network in the near future, in a fashion similar to *weather forecasting*. As another example, when the defender is able to accurately observe the cybersecurity state of each computer in a network (i.e., *full information*), the data may be leveraged to build models to predict the evolution of the cybersecurity states of those computers, which would enable *predictive* cybersecurity situational awareness. When the defender is able to accurately observe the cybersecurity state of *some*, but not *all*, of the computers in a network (i.e., *partial information*), the data may be leveraged to build models to infer the cybersecurity states of the other computers that are not directly observed. In order to achieve the tasks mentioned above, this level of abstraction would need to deal with the extraction of patterns exhibited by observed data (e.g., evolution of attack tactics).

The data-level abstraction also deals with the assumptions that are made by higher-level cybersecurity models or theories. In order to validate or invalidate such assumptions, we need to specify the kinds of data that must be collected. When the collection of such data is impossible or infeasible in the real world, this level of abstraction would need to deal with the design of experiments to generate and collect the desired data. The importance of cybersecurity experimentation has not been widely recognized.

The data-level abstraction further deals with the evolution of application- or mission-specific data items. This leads to the notion of *mission-data dynamics*, which describes the evolution of data items that are needed by, or relevant to, the mission in question. The dynamics allows to quantify, among other things, the dynamic or time-dependent trustworthiness of the data items because their trustworthiness evolves with time (e.g., owing to successful attacks during a period of time). This naturally bridges the semantic gaps between the applications of Cybersecurity Dynamics in risk quantification and decision-making (e.g., assessing the damage caused by successful attacks, or quantifying the confidence or uncertainty of a decision) and the underlying building-blocks.

2.2.4 What Is the Knowledge-level Abstraction? The knowledge-level of abstraction deals with the existence of laws that govern the evolution of the various kinds of dynamics and the identification of them when they indeed exist. At a conceptual level, laws are sought with respect to a *class* of random variables, stochastic processes, or dynamical systems. For example, a law may be derived from a Cybersecurity Dynamics model with respect to its component variables, which may follow certain statistical distributions, meaning that the law is applicable regardless of the specific instances of these random variables; a law may be derived from a Cybersecurity Dynamics model with respect to the properties of the parameters as long as they have certain properties (e.g., time-independent vs. time-dependent but with a certain property), meaning that the law is applicable regardless of the specific values of the model parameters as long as these properties are satisfied.

It is important to recognize the difference between the *laws* that are obtained at the knowledge-level abstraction and the *patterns* that are extracted at the data-level abstraction: The former are *data-independent* (or *data-agnostic*) and therefore more broadly applicable than the latter, which are *data-specific*. This is so because data-level abstractions are typically driven by specific observations of a certain kind of cybersecurity dynamics, which may be seen as *instances* of random variables, stochastic processes, or dynamical systems. This means that whatever patterns that may be extracted from, or whatever predictions that may be made for, a particular dataset are *only* applicable to that dataset. That is, the patterns may not be generalized to other instances of the random variable, stochastic process or dynamical systems.

3 THE CYBERSECURITY DYNAMICS LANDSCAPE

The Cybersecurity Dynamics way of thinking described above will lead to a systematic body of knowledge corresponding to the four levels of abstractions at the Dynamics Layer. In what follows we discuss the landscape with respect to the aforementioned way of thinking through the lens of *what* and *how* to model.

3.1 Coping with Dynamics Phenomena

As mentioned above, the dynamics concept is equally applicable to any cyber-physical-human system, including traditional enterprise IT infrastructures, cloud/edge/end computing systems, cyber physical systems, and cyber-physical-human systems. In order to model such dynamics, the *time* dimension must be adequately accommodated. In what follows we discuss how to achieve this with respect to networks, users, attacks, defenses, and cybersecurity states.

3.1.1 Coping with Network Dynamics. Networks evolve over time for many reasons, such as: (i) new computers may connect to, or be removed from, an enterprise network at any point in time; (ii) new devices may be added to, or removed from, a tactical network; (iii) employees' personal devices may be allowed to connect to an enterprise network in the wake of work-from-home policy in response to the COVID-19 pandemic; (iv) networking topology may evolve with time; and (v) the security policy enforced by a network may evolve with time. All these evolving networks can be modeled as *time-dependent* graphs $G(t) = (V(t), E(t))$, where $V(t)$ is the node or vertex set at time t and $E(t)$ is the edge set at time t.

One important issue is what $V(t)$ and $E(t)$ represent in the model, in order to robustly construct the elements of $G(t)$ from a cybersecurity standpoint. This of course depends on several factors, such as the cybersecurity problem that is under study, which would mandate an appropriate *granularity*. At a coarse granularity, each node (vertex) $v \in V(t)$ may represent a sub-network (e.g., treating a Department network as a node) and the edges in $E(t)$ may represent the communications between these sub-networks. At a finer granularity, each node (vertex) $v \in V(t)$ may present a computer or device. In this case, one edge may represent the wired or wireless link between two nodes (i.e., modeling the physical network), but there are other (perhaps more interesting) relationships that can be accommodated (as what will be discussed later). At an even finer granularity, each node (vertex) $v \in V(t)$ may represent a software component of interest (e.g., software application, operating system), an edge between two software components running on a computer may represent the caller-callee relation between them,

and an edge between two software components running on different computers may represent the communication relation between them (e.g., a browser communicates with a webserver) [8, 9]. With finer granularity, lower-level (i.e., more specific) information can be incorporated into $G(t)$. For example, *privilege escalation* can be accommodated at the preceding "finer" granularity, but cannot be described at the "coarse" granularity.

We hope to stress that even though it is tempting to use $G(t) = (V(t), E(t))$ to model dynamic *physical* networks, it is more important to use $G(t) = (V(t), E(t))$ to model what may be called *attack-defense structures*, where $(u, v) \in E(t)$ means that u can wage direct attacks against v (i.e., u is not prohibited by the network security policy from communicating with v). This means that $(u, v) \in E(t)$ often corresponds to a *communication path*, rather than a wired or wireless *link* in a physical network. Enforcing a security policy to restrict node connectivity (i.e., which nodes can communicate with which other nodes) is an important preventive defense mechanism [8, 9, 115, 129]. Intuitively, this policy corresponds to what has been widely employed in the real world: There are many physical facilities in the real world that cannot be accessed without being explicitly authorized in advance (e.g., airports).

At an appropriate granularity, each node $v \in V(t)$ may have some vulnerability at time t. It is important to model the evolution of vulnerabilities. For example, a vulnerability is discovered at time t_1 and patched at time t_2, which indicates when the attacker can and cannot exploit the vulnerability. It is equally important to model the evolution of v's susceptibility to attacks at different times, which may be caused by one or more weakness from the cyber domain (e.g., vulnerabilities in v's software), the physical domain (e.g., the susceptibility of v to being unplugged), or the human domain (e.g., the susceptibility of the user of v to being socially engineered). Moreover, a node $v \in V(t)$ often employs some defense mechanisms, such as anti-malware tool or host-based intrusion prevention or detection. These aspects can be readily described by associating parameters to the nodes. Furthermore, the defender often employs some network-based defense mechanisms (e.g., network-based intrusion prevention or detection). This can be readily described by associating parameters or "weights" to the edges. Putting all these together, we can naturally extend the time-dependent attack-defense structure $G(t) = (V(t), E(t))$ to $G(t) = (V(t), E(t), W_V(t), W_E(t))$, where $W_V(t)$ and $W_E(t)$ respectively describe the cybersecurity properties associated to the nodes and edges.

3.1.2 Coping with User Dynamics. Users evolve over time because a new employee may be hired at time t_1 and an employee may leave at time t_2. We may use $U(t)$ to denote the set of users at time t with respect to a network of interest. This is important because for a user $u \in U(t_1), U(t_2)$, u's susceptibility to social engineering cyber attacks at time t_2 can be different from its counterpart at time t_1. Modeling these susceptibility dynamics is important in order to characterize the evolution in terms of a user's awareness of, and skills in countering, these attacks [86].

In order to model the susceptibility of node $v \in V(t)$ to cyber attacks that may be incurred by its user's vulnerability to social engineering cyber attacks, it is important to model which users have an impact on which computers. For this purpose, we can map the users in $U(t)$ to the nodes in $V(t)$ to describe which user is authorized to use which computer. This mapping can be represented by a bipartite graph, $G_{\text{usage}}(t) = (U(t), E_{\text{usage}}(t))$, where $E_{\text{usage}}(t) \subseteq U(t) \times V(t)$ represents which user in $U(t)$ is authorized to use which computer in $V(t)$.

3.1.3 Coping with Attack Dynamics. Cyber attacks evolve with time. The evolution of cyber attacks can be modeled from the following perspectives. First, cyber attacks are waged by evolving sets of *attackers* that may have different sets of attack skills and may exhibit different human factors (e.g., cognitive capabilities). We can use $\mathsf{A}_1(t)$ to describe the set of attackers that conduct attacks at time t. Second, attackers also use evolving sets of *exploits* because new exploits may emerge and old exploits may become obsolete. We can denote by $\mathsf{A}_2(t)$ the set of exploits that are available to the attackers at time t. Third, attackers may use evolving sets of *strategies*, such as the Cyber Kill Chain [43] or the Att&CK model [93], to guide their attacks. We can denote by $\mathsf{A}_3(t)$ the set of strategies that are available to the attackers at time t. Fourth, attackers may use evolving sets of *tactics*, such as leveraging malicious websites vs. malicious emails. We can use $\mathsf{A}_4(t)$ to denote the set of attack tactics that are available to the attackers at time t. Fifth, attackers may use evolving sets of *orientation* algorithms in their Observe-Orient-Decide-Act (OODA) loops [5]. We can use $\mathsf{A}_5(t)$ to describe the set of orientation algorithms that are available to the attackers at time t. Sixth, attackers may use evolving sets of *decision-making* algorithms in their OODA loops (e.g., for minimizing the probability that the ongoing attacks are detected by the defenders). We can use $\mathsf{A}_6(t)$ to denote the set of decision-making algorithms that are available to the attackers at time t.

The preceding distinctions between each $\mathsf{A}_i(t)$ pave a way for modeling the discrepancies imposed by different combinations of them. For example, the distinction between $\mathsf{A}_1(t)$ and $\mathsf{A}_2(t)$ paves a way for modeling the *human-tool* cooperation because different attackers teamed with the same attack tool may lead to different outcomes or effectiveness; similarly, different attack tools teamed with the same attacker may lead to different outcomes or effectiveness. In summary, attacks at time t can be described by $\mathsf{A}(t) = (\mathsf{A}_1(t), \ldots, \mathsf{A}_6(t))$.

3.1.4 Coping with Defense Dynamics. Cyber defenses evolve with time. The evolution of cyber defenses can be modelled from the following perspectives. First, cyber defenses are conducted by evolving sets of *defenders* that may have different set of defense skills because new defenders may be hired and some defenders may leave. We can use $\mathsf{D}_1(t)$ to describe the set of defenders that conduct cyber defense at time t. Second, defenders may use evolving sets of defense *tools* (e.g., using new tools or updating the attack signatures or detection models of certain tools). We can use $\mathsf{D}_2(t)$ to describe the set of defense tools that are employed at time t. Third, defenders may use evolving sets of *strategies*, such as NIST's Cybersecurity Framework [78] to guide their defenses. We can use $\mathsf{D}_3(t)$ to describe the set of defense strategies that are employed at time t. Fourth, defenders may use evolving sets of *tactics*, such as leveraging preventive defense (e.g., enforcing access controls), leveraging reactive defense (e.g., employing attack detectors), leveraging adaptive defense (e.g., dynamically adjusting defense configurations or postures), leveraging proactive defense (e.g., employing moving

target defense even if attacks are not detected), leveraging active defense (e.g., employing autonomous agents to fight against attackers' autonomous agents), and leveraging deception (e.g., employing honeypot-like defense instruments). We can use $\mathsf{D}_4(t)$ to denote the set of defense tactics that are available to the attackers at time t. Fifth, defenders may use evolving sets of *orientation* algorithms in their OODA loops [5]. We can use $\mathsf{D}_5(t)$ to describe the set of orientation algorithms employed by the defenders at time t. Sixth, defenders may use evolving sets of *decision-making* algorithm in their OODA loops (e.g., minimizing the cost incurred by the dynamic orchestration of cyber defense). We can use $\mathsf{D}_6(t)$ to denote the set of decision-making algorithms that are employed by the defenders at time t.

The preceding distinctions between each $\mathsf{D}_i(t)$ pave a way for modeling the discrepancies imposed by different combinations of them. For example, the distinction between $\mathsf{D}_1(t)$ and $\mathsf{D}_t(t)$ paves a way for modeling the *human-tool* cooperation because different defenders teamed with the same defense tool may lead to different outcomes or effectiveness; similarly, different defense tools teamed with the same defender may lead to different outcomes or effectiveness. In summary, defenses at time t can be described by $\mathsf{D}(t) = (\mathsf{D}_1(t), \ldots, \mathsf{D}_6(t))$.

3.1.5 Coping with Cybersecurity State Dynamics. Intuitively, the cybersecurity state at a specific point in time reflect the particular situation. Moreover, cybersecurity states evolve over time because of the dynamic attack-defense interactions in networks [110–112]. Cybersecurity states can be defined based on the network dynamics, especially the cybersecurity states of nodes $V(t)$ in the attack-defense structure $G(t) = (V(t), E(t))$. Since network dynamics can be defined at multiple granularities, cybersecurity states can be defined correspondingly. At the granularity where each node $v \in V(t)$ represents a computer, the cybersecurity state of v at time t can be *secure* (i.e., v is neither exposed to the attacker nor compromised by the attacker), *exposed* (i.e., v is known to the attacker), or *compromised* (i.e., v is under the control of the attacker). The same can be defined when v represents a software component, such as the aforementioned distinction between application vs. operating system, where a computer is modeled as multiple nodes in $G(t)$. This effectively leads to the notion of multilayer networks, where an application is compromised does not necessarily mean the underlying operating system is compromised.

Cybersecurity states can also be defined based on user dynamics $U(t)$, attack dynamics $\mathsf{A}(t)$ and defense dynamics $\mathsf{D}(t)$. For example, the cybersecurity state corresponding to $U(t)$ can be defined to measure the users' susceptibility to social engineering attacks and insider threats at time t. The cybersecurity state corresponding to $\mathsf{A}(t)$ can be defined to measure the attack capabilities or threats at time t (e.g., competence of attack strategies). The cybersecurity state corresponding to $\mathsf{D}(t)$ can be defined to measure the defense capabilities at time t (e.g., competence of defense tactics).

3.2 Coping with the Metrics-Level Abstraction

The importance of cybersecurity metrics cannot be overstated [16, 74, 75, 82, 85, 89, 92]. This level of abstraction may include multiple sub-levels of abstractions.

3.2.1 The Goal. The ultimate goal at this level of abstraction is to build a systematic theory of *cybersecurity metrics*, broadly defined to include resilience metrics and agility metrics [15]. At a high level, achieving this ultimate goal requires accomplishing a range of tasks related to the various needs at the dynamics layer and the application layer as illustrated in Figure 1. We advocate distinguishing *atomic* metrics from *composite* metrics. An atomic metric is one that can be directly measured from a supporting dataset. A composite metric is one that may not be directly measured, because it is infeasible or too costly to do so, but instead can be derived from a *computational* procedure that takes some atomic metrics as the input. In this paper, a *computational* procedure can be a mathematical function, algorithm, statistical method, or machine learning model.

In order to describe and characterize the *network dynamics*, namely the evolution of $G(t) = (V(t), E(t))$ with time t, one needs to determine the *granularity* because it determines what $v \in V(t)$ represents. Owing to the complexity of the cybersecurity problem, multiple granularities might need to be considered, such as subnetwork vs. computer vs. software component as mentioned above. Correspondingly, $e \in E(t)$ has different meanings at different granularities. In principle, metrics need to be defined to describe the properties of $v \in V(t)$ and $e \in E(t)$. When the granularity is that each $v \in V(t)$ represents a computer, metrics can be defined to describe (i) v's vulnerabilities to attacks and (ii) v's defense capabilities. The preceding (i) may be defined as a composite metric that can be derived from the following two atomic metrics: one describes the vulnerability of the software stack running on v and the other describes its user's vulnerability to social engineering cyber attacks or insider threat. The preceding (ii) may reflect the host-based defense employed on v. Correspondingly, metrics can also be defined to be associated with $e \in E(t)$ and possibly reflect the network-based and/or host-based defense capabilities.

In order to describe and characterize the *user dynamics* $U(t)$ over time t, one needs to define metrics to measure the users' vulnerabilities to to social engineering cyber attacks and insider threats (i.e., how likely a user is an insider attacker). At any point in time t, one metric may be defined to describe the likelihood or probability that $u \in U(t)$ is vulnerable to social engineering cyber attacks, and another metric may be defined to describe the likelihood or probability that $u \in U(t)$ is an insider. These metrics would be composite metrics. Recently, a cybersecurity cognitive psychology framework [86] has been proposed to investigate how the former metric may be derived from factors reflecting a user's cognitive factors. On the other hand, the latter metric may be derived from factors reflecting a user's behavior.

In order to describe and characterize the *attack dynamics* $\mathsf{A}(t) = (\mathsf{A}_1(t), \ldots, \mathsf{A}_6(t))$ over time t, we can define metrics to measure the following: (i) the attackers' cognitive or psychological capabilities, which reflect their susceptibility or likelihood to fall trap into the defender's deception schemes; (ii) the set and/or capabilities of the exploits that are available to the attackers; (iii) the competency of the attack strategies that are available to the attackers; (iv) the competency of the attack tactics that are available to the attackers; (v) the competency of the orientation algorithms that are available to the attackers; (vi) the competency of the decision-making algorithms that are available to the attackers.

In order to describe and characterize the *defense dynamics* $\mathrm{D}(t) = (\mathrm{D}_1(t), \ldots, \mathrm{D}_6(t))$ over time t, we can define metrics to measure the following: (i) the defenders' cognitive or psychological capabilities, which reflect their susceptibility or likelihood to fall into the trap of the attackers' deception schemes; (ii) the set and/or capabilities of the defense tools that are available to the defender; (iii) the competency of the defense strategies that are available to the defender; (iv) the competency of the defense tactics that are available to the defender; (v) the competency of the orientation algorithms that are available to the defender; (vi) the competency of the decision-making algorithms that are available to the defender.

In order to describe and characterize the *cybersecurity state dynamics*, namely the evolution of the cybersecurity state of a network over time, we can measure the state of $v \in V(t)$. At any point in time t, v may be in one, and only one, of multiple possible states of interest, such as the aforementioned *secure* vs. *exposed* vs. *compromised* state. Alternatively, the cybersecurity state of a node v can be approximated by the probability that the node is in the a specific state at time t. It may be tempting to treat the cybersecurity state metrics as atomic metrics because in principle, these states can be directly measured by examining each node at any time t. However, this measurement procedure is not feasible because we often deal with large networks, which cannot be exhaustively measured with sufficient precision. This explains why it is important to treat them as composite metrics. This can be achieved by deriving them as the outcome of attack-defense interactions on $G(t)$. This view makes it possible to identify some of the laws that govern the evolution of the cybersecurity state.

3.2.2 The State-of-the-Art. In principle, the evolution of $G(t) = (V(t), E(t))$ can be described as a stochastic process. This turns out to be a difficult task, explaining why most existing studies make simplifying assumptions, such as assuming time-independent $G(t) = (V(t), E(t))$, namely $G = (V, E)$ regardless of t, or assuming time-independent $V(t)$ and special kinds of (rather than arbitrarily) time-dependent $V(t)$, namely $G(t) = (V, E(t))$. There are some recent progresses on cybersecurity metrics [8, 9, 13–15, 41, 68, 76, 82, 85, 97, 98, 127]. A number of metrics have been defined to measure the properties associated to $v \in V(t)$ and $e \in E(t)$ in the network dynamics, the properties associated to $u \in U(t)$ in the user dynamics, the properties of the attackers in the context of the attack dynamics, the properties of the defenders in the context of the defense dynamics, and the cybersecurity state. However, much research remains to be done because it is still unknown what metrics are *necessary* and *sufficient* in order to achieve quantitative cyber risk management and cyber defense decision-making [82].

3.2.3 Future Research Directions. The most outstanding open problem is to define a suite of cybersecurity metrics that are both necessary and sufficient to achieve quantitative cyber risk management and cyber defense decision-making. We anticipate that achieving this ultimate goal will take the effort of the entire community for many years to come. We believe that significant progress can be made in the following aspects. One aspect is to define a spectrum of metrics to serve the needs of the data-level abstraction, the knowledge-level abstraction, and the application layer. For example, it is important to define metrics to measure attack capabilities of exploits and defense capabilities of defense tools, not only from a building-block point of view (i.e., considering them in an isolated network or computer) but also from a whole-system point of view (i.e., considering them in an open network). Moreover, the measurement of these metrics needs to be systematically investigated. It is also important to identify the criteria that can be used to distinguish the *good* metrics and the *poor* ones.

3.3 Coping with the Data-Level Abstraction

We cannot measure cybersecurity metrics without data and we cannot validate/invalidate assumptions without data. At the data-level of abstraction, we can build models to describe the evolution of the cybersecurity phenomena exhibited by the data. This level of abstraction may include multiple sub-levels of abstractions.

3.3.1 The Goal. The ultimate goal at this level of abstraction is to build a systematic theory of *cybersecurity data science*. At a high level, achieving this ultimate goal would require accomplishing a range of tasks, including the following. The first task is to investigate, in order to measure the metrics that are defined at the metrics-level abstraction, what kinds of data need to be collected and how to collect that data. In the case that some kinds of data may be hard or costly to collect, this level of abstraction should give feedback to the metrics-level of abstraction to suggest the definition and use of alternate metrics that can be easily collected. The second task is to investigate, in order to validate/invalidate the assumptions that are made at higher levels of abstractions (i.e., the knowledge-level abstraction and the application layer), what kinds of data need to be collected. When some of the assumptions made by the higher levels of abstractions are hard or costly to validate/invalidate, the present level of abstraction needs to suggest alternate assumptions to the higher level ones. This task may also include the design of cyber attack-defense experiments. The third task is to extract data-driven patterns and forecast/predict their evolution. The data-driven patterns often lead to insights that can not only deepen our understanding but also serve as hints for refining the metrics-level and knowledge-level abstractions as well as the application-layer models.

3.3.2 The State-of-the-Art. There is some recent progress at the data-level of abstraction. Towards measuring fine-grained attack-defense structure $G(t)$, [8, 9] have investigated the finer granularity that $v \in V(t)$ represents a software component (e.g., application, library, or operating system). Towards measuring the residual vulnerability of software programs, there have been studies on the capabilities and limitations of software vulnerability detectors, including similarity-based detectors [49, 59] and pattern-based detectors [33, 58, 60, 61, 72, 120, 121, 130]. Towards measuring attack capabilities, there have been studies on measuring the capabilities of malicious attacks against malicious website detectors (see, e.g., [102, 103]) and more recently on measuring the capabilities of malware detector against intelligent adversarial examples [6, 11, 54, 55, 99, 119]. Towards measuring defense capabilities, existing measurement methods often depend on the availability of ground truth data, which may not be available in the real world. There have been studies on alleviating this assumption [7, 24, 45]. Towards extracting data-driven patterns and forecasting

cyber threats at aggregate levels, there has been significant progress [12, 25, 83, 84, 105, 106, 124–126].

3.3.3 Future Research Directions. There are many outstanding open problems at the data level of abstraction, especially identifying the appropriate kinds of data to connect the desired metrics and the real-world networks (including their cybersecurity architectures and mechanisms). In what follows we discuss some of them.

The first outstanding open problem is to systematically characterize what kinds of data are necessary and sufficient for cyber defense purposes, including quantitative cyber risk management and decision-making. For example, in order to measure the capabilities of preventive defense tools or mechanisms, we might need to consider the kinds of exploits or attacks that may be prevented by those preventive defense tools or mechanisms (e.g., the trustworthiness of digital signatures [10, 18, 20, 22, 34, 37, 39, 44, 66, 79, 101, 113, 118], the trustworthiness of cryptographic services [19, 21, 23, 108], and the trustworthiness of security services [4, 27, 28]). In order to measure the capabilities of reactive defense tools or mechanisms, we might need to consider their detection capabilities [7, 24, 69, 70].

The second outstanding open problem is to cope with the lack of data incurred by the fact that data (can be collected but) cannot be shared to the research community for privacy and/or policy reasons. Although there have been efforts at sharing data (e.g., the DHS-sponsored IMPACT project), much greater efforts need to be made. In particular, we need to characterize how the available data can be *pieced together* to formulate a more holistic view of the dynamic situation and what gaps need to be bridged.

The third outstanding open problem is to characterize what can and cannot be predicted, to what extent, and at what level of certainty. One challenge is to deal with the inherent data sparsity that can be encountered at the data-level of abstraction. This happens when dealing with infrequent events which incur large damages. One concrete example is enterprise-level *data breach* incidents, which only happen to an enterprise once or twice over the span of many years. A significant progress in this direction is very recently made in [26]. Another example is to understand the extent to which the emergence of new or 0-day vulnerabilities might be predictable.

3.4 Coping with the Knowledge-Level Abstraction

This level of abstraction goes beyond the data-level abstraction. This can be seen from the following simple example: The data collected in the real world can be seen as a *particular instance* of a random variable or stochastic process, meaning that the patterns extracted from a particular dataset may not be generalized to the other instances of the same random variable or stochastic process. This is because it is often impossible to collect data corresponding to *multiple instances* of the same random variable or stochastic process, which is a reality in sharp contrast to what is taught in statistics textbooks. On the other hand, the knowledge-level abstraction aims to create a body of knowledge that is, in this example, applicable to the random variable or stochastic process in question as along as they possess some properties (i.e., a certain distribution associated to the random variable). This level of abstraction may include multiple sub-levels of abstractions.

3.4.1 The Goal. The ultimate goal at this level of abstraction is to build a systematic (and unified) theory of *first-principle* or *theoretical cybersecurity science*. The notion of *first-principle* in the context means "*making as-weak-as-possible assumptions, building as-simple-as-possible models with cybersecurity meanings, and using as-few-as-possible parameters*" [112]. This level of abstraction aims to discover the knowledge that cannot be directly extracted from cybersecurity data, but has to be inferred or reasoned by considering the attack-defense-user interactions as a whole. A kind of knowledge of particular interest is the laws that govern the evolution of the network dynamics, the user dynamics, the attack dynamics, the defense dynamics, and the cybersecurity state dynamics. A systematic cybersecurity dynamics model would need to accommodate: (i) the network dynamics $G(t)$ and the accompanying metrics that can serve as model parameters; (ii) the user dynamics $U(t)$ and the accompanying metrics that can serve as model parameters; (iii) the attack dynamics $\mathsf{A}(t) = (\mathsf{A}_1(t), \ldots, \mathsf{A}_6(t))$ and the accompanying metrics that can serve as model parameters; (iv) the defense dynamics $\mathsf{D}(t) = (\mathsf{D}_1(t), \ldots, \mathsf{D}_6(t))$ and the accompanying metrics that can serve as model parameters; and (v) the cybersecurity state dynamics $S(t)$. In principle, the evolution of $S(t)$ is driven by the interactions between attacks $\mathsf{A}(t)$ and $\mathsf{D}(t)$ on network $G(t)$ with humans $U(t)$ in the loop. A particular kind of interactions can be represented as a mathematical function, denoted by f, namely

$$S(t) = f(G(t), U(t), \mathsf{A}(t), \mathsf{D}(t)). \tag{1}$$

Different kinds of attacker-defender-user interactions lead to different kinds of f. Identifying the appropriate f and characterizing its implication are two of the research tasks at this level of abstraction.

3.4.2 State-of-the-Art. Several families (or classes) of first-principle cybersecurity dynamics models have been investigated. The first family is known as *preventive and reactive cybersecurity dynamics* [8, 9, 30, 57, 64, 104, 107, 115, 117, 122, 123, 129]. These models accommodate two kinds of defenses: *preventive defenses*, which include both host- and network-based preventive defenses (e.g., enforcing various kinds of access control policies), and *reactive defenses*, which include both host- and network-based reactive defenses (e.g., anti-malware tools that aim to detect compromised computers and clean them up). Two kinds of attacks have been investigated in this context: *push-based* ones, which model attacks such as malware spreading, and *pull-based* ones, which model attacks such as *drive-by downloads*.

This is, relatively speaking, the most understood family of cybersecurity dynamics models since its introduction in [56]. It took 10 years for researchers to attain a complete characterization the dynamics [129], by rigorously proving the global stability of this dynamics in the *entire* parameter universe. This means that the dynamics always converges to a unique equilibrium regardless of the initial state or when the defender starts to observe/monitor the dynamics. A more recent result [63] shows that this global convergence property holds for a more general family of dynamics that accommodates [56] and another dynamics [77, 96] as special cases. An even more recent result is a characterization of the (so far) weakest condition under which this dynamics is globally attractive to a trajectory (rather than an equilibrium) [36].

It is worth mentioning that the theoretical studies mentioned above have important applications. One application is to guide the design of practical statistical methods for *estimating* the network-wide cybersecurity equilibrium state *without* knowing the values of the model parameters [115, 129]. Another application is the identification of the following fundamental limitation of preventive and reactive cyber defenses: the attack effect is amplified by the network $G(t)$ but the defense effect is not. This suggests that in order to effectively defend against the attacks, we need to reduce the amplification effect and/or use other kinds of cyber defenses.

There are other families of cybersecurity dynamics, including: adaptive cybersecurity dynamics [17, 116], where the defender can promptly adapt the defense posture including $\mathrm{D}(t)$ and possibly $G(t)$ and $U(t)$; proactive cybersecurity dynamics [35], where the defender can proactively adjust the defense posture; and active cybersecurity dynamics [114, 128], where the defender can leverage autonomous software agents to fight cyber attacks. Active cybersecurity dynamics are closely related to what is called *autonomous defense* by other researchers [51–53, 65, 94, 95]. However, it has been shown that active cyber defense can make cybersecurity unmanageable because it can induce some chaotic phenomena [128].

3.4.3 Future Research Directions. There are many open problems at this level of abstraction. These open problems are associated with a range of inherent technical barriers, which have been discussed in [112] and are refined as follows.

- The *scalability* barrier says that native stochastic-process models for cybersecurity dynamics would immediately encounter the *state-space explosion* problem, which has motivated the alternative of modeling the probability that a node $v \in V(t)$ is in a certain state (e.g., compromised) rather than the deterministic assertion that v is indeed in a specific state.
- The *nonlinearity* barrier in inherent to the complex attacker-defense-user interactions on complex networks that lead to the abstraction of $G(t)$. Experiments need to be conducted to validate which kinds of nonlinearity are relevant.
- The *dependence* barrier [110] is also inherent to the complex attacker-defender-user interactions, such as coordinated attacks and coordinated defenses [109]. Although progresses have been made towards tackling dependence [17, 104, 107], much more studies remain to be done. It is worth mentioning that while dependence can render theoretical cybersecurity models difficult to analyze, it can be leveraged to achieve data-driven prediction despite sparse data [26].
- The *structural dynamics* barrier is encountered when extracting the attack-defense structure $G(t)$ corresponding to a given, but dynamically evolving, network. One particular challenge is to automate the extraction of $G(t)$ from a given network while accommodating its security policies, which are often very complex and poorly documented.
- The *transient behavior* barrier is imposed when characterizing the behavior of a cybersecurity dynamics model *before* the dynamics converge to an equilibrium, if it does at all. Results in this regard often deal with *asymptotic* behaviors with respect to sufficiently large time t (or $t \to \infty$ in the mathematical terminology). However, cybersecurity is such a field that the behavior before converging to the equilibrium is perhaps even more important. One solution concept is to apply bounds to the metrics of interest. Another approach is to leverage the statistical way of thinking.
- The *uncertainty* barrier is manifested by several aspects: *model uncertainty*, which corresponds to the degree of faithfulness of a model to the reality that is being modeled; *parameter uncertainty*, which corresponds to the uncertainty in the model parameters (e.g., measurement errors); *information uncertainty*, which reflects that only partial information, rather than full information, is known. These uncertainties have not been systematically investigated.
- The *human factor* barrier is related to measuring users' susceptibility to social engineering attacks, the attacker's susceptibility to defender's deception schemes, and the defender's susceptibility to the attacker's deception schemes. Recently, a framework for quantifying human's cognitive capability against social engineering cyber attacks have been proposed [86]. The framework aims to quantify the behavior of a human, such as a user or defender, with unique short-term and long-term cognition factors in response to social engineering cyber attacks. The framework sheds light on an emerging sub-field that may be called *Cybersecurity Cognitive Psychology*, which aims to tailor the Cognitive Psychology principles to the cybersecurity domain. One particular aspect of the human factor barrier is the *deception* barrier, which is manifested by the limitation of human cognition and the availability of data. For example, an attacker can intentionally deceive a defender into believing that the network is in a certain state so as to cause the defender to make wrong decisions. As a double-edged sword, deception can be leveraged by the defender to deceive the attacker. One challenge is that the cognitive aspect of human deception or human cognition itself is little understood.

3.5 Killer Applications of the Cybersecurity Dynamics Framework

We envision two example killer applications of the Cybersecurity Dynamics framework: *quantitative cyber risk management* and *realizing cyber defense OODA loop*.

3.5.1 Application 1: Quantitative Cybersecurity Risk Management. Quantitative cybersecurity risk management is an important application that is encountered by practitioners. One particular risk management task is to decide how to invest in cybersecurity in a cost-effective, if not optimal, fashion. This can be manifested by the employment of different defense postures at time t_0, which can be modeled by $G(t)$ vs. $G'(t)$, namely the attack-defense structure as it might change with a prospective defense investment. Using the notations defined above, this will allow the defender to contrast $S(t) = f(G(t), U(t), \mathrm{A}(t), \mathrm{D}(t))$ and $S'(t) = f(G'(t), U(t), \mathrm{A}(t), \mathrm{D}(t))$ for $t > t_0$ according to Eq. (1), and therefore select the better attack-defense structure. Similarly, the defender may need to choose between different training methods for reducing users' susceptibility to social engineering cyber attacks, leading to the contrast $U(t)$ vs. $U'(t)$. Correspondingly, this will allow the defender to contrast $S(t) = f(G(t), U(t), \mathrm{A}(t), \mathrm{D}(t))$ and $S'(t) = f(G(t), U'(t), \mathrm{A}(t),$

$\mathrm{D}(t)$) for $t > t_0$ according to Eq. (1), and therefore select the more effective methods to train the users.

The defender may need to choose between different defenders to hire, leading to the contrast $\mathrm{D}_1(t)$ vs. $\mathrm{D}'_1(t)$; the defender may need to choose between different defense tools, leading to the contrast $\mathrm{D}_2(t)$ vs. $\mathrm{D}'_2(t)$; the defender may need to choose between different defense strategies, leading to the contrast $\mathrm{D}_3(t)$ vs. $\mathrm{D}'_3(t)$; the defender may need to choose between different defense tactics, leading to the contrast $\mathrm{D}_4(t)$ vs. $\mathrm{D}'_4(t)$; the defender may need to choose between different defense orientation algorithms, leading to the contrast $\mathrm{D}_5(t)$ vs. $\mathrm{D}'_5(t)$; the defender may need to choose between different decision-making, leading to the contrast $\mathrm{D}_6(t)$ vs. $\mathrm{D}'_6(t)$. In any case, the choices will lead to the contrast $\mathrm{D}(t)$ vs. $\mathrm{D}'(t)$, which can respectively lead to $S(t) = f(G(t), U(t), \mathrm{A}(t), \mathrm{D}(t))$ vs. $S'(t) = f(G(t), U(t), \mathrm{A}(t), \mathrm{D}'(t))$ for $T > t$ according to Eq. (1). By contrasting $S(t)$ and $S'(t)$ for $t > t_0$, the defender can make a principled and quantitative decision in selecting $\mathrm{D}(t)$ or $\mathrm{D}'(t)$.

3.5.2 Application 2: Realizing Autonomous, Agile and Quantitative Cyber Defense OODA Loop. The OODA loop [5] formulates a systematic way of thinking in combat operation processes. As highlighted in Figure 2, the OODA loop can be equally applied to cyberspace. In this context, *observe* corresponds to monitoring the situation of a network; *orient* corresponds to analyzing the evolution of the situation within a time constraint; *decide* corresponds to making decisions to achieve optimal cyber maneuvers within a time constraint; and *act* corresponds to executing the select cyber maneuvers. In particular, a core function of the *Agile Cyber Defense Orientation (CDO) Engine* is to achieve *predictive* cybersecurity situational awareness within a time constraint, and a core function of the *Agile Cyber Defense Command and Control (CCDC) Engine* is the decision-making module.

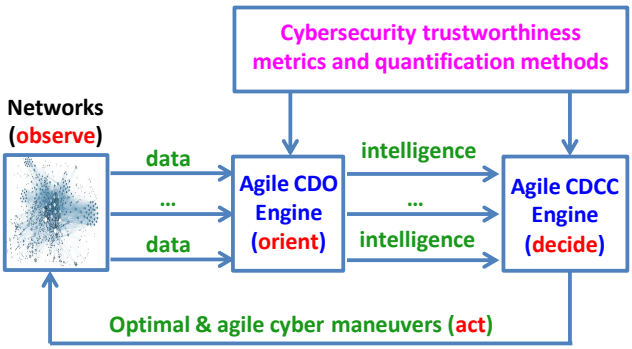

Figure 2: The Cybersecurity Dynamics view of cyber defense operations.

Figure 3 illustrates one scenario of decision-making based on *predictive* Cybersecurity Dynamics models, including both first-principle cybersecurity models and data-driven models. The key insight into this decision-making function is that it leverages not only the present data (reflecting the current situation) but also the past and predicted future data (reflecting respectively the past and future situations) for decision-making. Moreover, the decision-making function may make a range of recommendations to a human decision-maker (if applicable), where each recommendation is

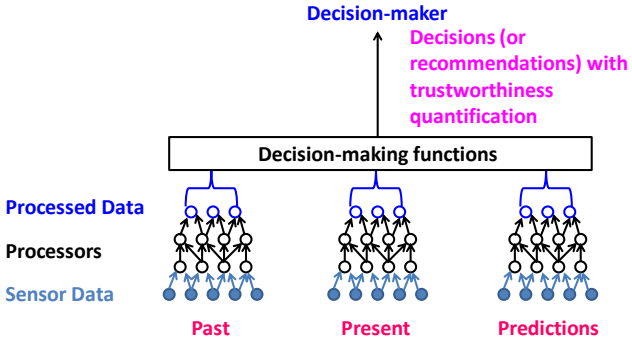

Figure 3: Illustration of the decision-making module of the CDCC based on *predictive* Cybersecurity Dynamics models.

accompanied with a quantified trustworthiness, reflecting the confidence of the decision-making algorithm given the various kinds of uncertainties that are inherent to the decision-making model and/or the data used in the decision-making process.

4 CONCLUSION

We have described the Cybersecurity Dynamics way of thinking, which highlights a paradigm shift from *time-independence* to *time-dependence*. The time-dependent nature of cybersecurity leads to the perspective of *dynamics*. We have described the Cybersecurity Dynamics landscape via a *dynamics layer* for characterizing cybersecurity and supporting various cybersecurity functions. We hope this paper will inspire many researchers to tackle the range of outstanding open problems discussed in the paper. We believe that these open problems, especially the technical barriers, are inherent, meaning that they must be tackled regardless of specific approaches.

ACKNOWLEDGMENTS

The author would like to thank Dr. Hamed Okhravi and Dr. Cliff Wang, the PC Co-Chairs of ACM MTD'2020, not only for their invitation for writing this paper but also for their suggestion of the term "Cybersecurity Dynamics landsacpe" which is incorporated into the title of the paper. The author would like to further thank Dr. Cliff Wang for his insightful comments on a manuscript of the paper, which led to, among other things, the notion of *mission-data dynamics* mentioned in the paper. The author would like to thank Huashan Chen and Eric Ficke for their comments and proofreading the paper. The author would like to thank his mentors for their encouragement, philosophical advice, and moral support over the past many years. The author would like to thank his many collaborators (including students) on this ambitious endeavor. The work is supported in part by ARO Grant #W911NF-17-1-0566 as well as NSF Grants #1814825 and #1736209.

REFERENCES

[1] R. Anderson and R. May. 1991. *Infectious Diseases of Humans.* Oxford University Press.
[2] N. Bailey. 1975. *The Mathematical Theory of Infectious Diseases and Its Applications.* 2nd Edition. Griffin, London.

[3] A. Barrat, M. Barthlemy, and A. Vespignani. 2008. *Dynamical Processes on Complex Networks.* Cambridge University Press.

[4] Nathaniel Boggs, Senyao Du, and SalvatoreJ. Stolfo. [n.d.]. Measuring Drive-by Download Defense in Depth. In *Proc. RAID'14*. 172–191.

[5] John Boyd. 28 June 1995. The Essence of Winning and Losing.

[6] N. Carlini and D. Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*. 39–57.

[7] J. Charlton, P. Du, J. Cho, and S. Xu. 2018. Measuring Relative Accuracy of Malware Detectors in the Absence of Ground Truth. In *IEEE MILCOM*.

[8] H. Chen, J. Cho, and S. Xu. 2018. Quantifying the security effectiveness of firewalls and DMZs. In *Proc. HoTSoS'2018.* 9:1–9:11.

[9] H. Chen, J. Cho, and S. Xu. 2018. Quantifying the security effectiveness of network diversity. In *Proc. HoTSoS'2018.* 24:1.

[10] H. Chen, M. Pendleton, L. Njilla, and S. Xu. 2020. A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses. *ACM Comput. Surv.* 53, 3 (2020), 67:1–67:43.

[11] L. Chen, S. Hou, Y. Ye, and S. Xu. 2018. DroidEye: Fortifying Security of Learning-based Classifier against Adversarial Android Malware Attacks. In *Proc. 2018 IEEE/ACM ASONAM*. 782–789.

[12] Y. Chen, Z. Huang, S. Xu, and Y. Lai. 2015. Spatiotemporal patterns and predictability of cyberattacks. *PLoS One* 10, 5 (05 2015), e0124472.

[13] Y. Cheng, J. Deng, J. Li, S. DeLoach, A. Singhal, and X. Ou. 2014. Metrics of Security. In *Cyber Defense and Situational Awareness*. 263–295.

[14] J. Cho, P. Hurley, and S. Xu. 2016. Metrics and Measurement of Trustworthy Systems. In *IEEE Military Communication Conference (MILCOM 2016)*.

[15] J. Cho, S. Xu, P. Hurley, M. Mackay, T. Benjamin, and M. Beaumont. 2019. STRAM: Measuring the Trustworthiness of Computer-Based Systems. *ACM Comput. Surv.* 51, 6 (2019), 128:1–128:47.

[16] INFOSEC Research Council. 2007. Hard Problem List. http://www.infosec-research.org/docs_public/20051130-IRC-HPL-FINAL.pdf.

[17] G. Da, M. Xu, and S. Xu. 2014. A New Approach to Modeling and Analyzing Security of Networked Systems. In *Proc. HotSoS'14.* 6:1–6:12.

[18] W. Dai, P. Parker, H. Jin, and S. Xu. 2012. Enhancing Data Trustworthiness via Assured Digital Signing. *IEEE TDSC* 9, 6 (2012), 838–851.

[19] Y. Desmedt and Y. Frankel. 1989. Threshold cryptosystems. In *Crypto*. 307–315.

[20] X. Ding, G. Tsudik, and S. Xu. 2004. Leak-Free Group Signatures with Immediate Revocation.. In *Proc. ICDCS*. 608–615.

[21] Y. Dodis, J. Katz, S. Xu, and M. Yung. 2002. Key-Insulated Public Key Cryptosystems. In *Proc. EUROCRYPT*. 65–82.

[22] Y. Dodis, J. Katz, S. Xu, and M. Yung. 2003. Strong Key-Insulated Signature Schemes.. In *Public Key Cryptography (PKC'03)*. 130–144.

[23] Y. Dodis, W. Luo, S. Xu, and M. Yung. 2012. Key-insulated symmetric key cryptography and mitigating attacks against cryptographic cloud software. In *Proc. ASIACCS'12*.

[24] P. Du, Z. Sun, H. Chen, J. H. Cho, and S. Xu. 2018. Statistical Estimation of Malware Detection Metrics in the Absence of Ground Truth. *IEEE T-IFS* 13, 12 (2018), 2965–2980.

[25] X. Fang, M. Xu, S. Xu, and P. Zhao. 2019. A deep learning framework for predicting cyber attacks rates. *EURASIP J. Information Security* 2019 (2019), 5.

[26] Z. Fang, M. Xu, S. Xu, and T. Hu. 2020. Framework for Predicting Data Breach Risk: Leveraging Dependence to Cope with Sparsity. *manuscript* (2020).

[27] E. Ficke, K. Schweitzer, R. Bateman, and S. Xu. 2018. Characterizing the Effectiveness of Network-Based Intrusion Detection Systems. In *MILCOM*. 76–81.

[28] E. Ficke, K. Schweitzer, R. Bateman, and S. Xu. 2019. Analyzing Root Causes of Intrusion Detection False-Negatives: Methodology and Case Study. In *MILCOM*.

[29] A. Ganesh, L. Massoulie, and D. Towsley. 2005. The Effect of Network Topology on the Spread of Epidemics. In *Proceedings of IEEE Infocom 2005*.

[30] Richard Garcia-Lebron, David J Myers, Shouhuai Xu, and Jie Sun. 2019. Node diversification in complex networks by decentralized colouring. *Journal of Complex Networks* 7, 4 (5 2019), 554–563.

[31] S. Goldwasser and S. Micali. 1982. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *ACM STOC*. 365–377.

[32] S. Goldwasser, S. Micali, and C. Rackoff. 1985. The knowledge complexity of interactive proof-systems. In *ACM STOC*. 291–304.

[33] G. Grieco, G. Grinblat, L. Uzal, S. Rawat, J. Feist, and L. Mounier. 2016. Toward large-scale vulnerability discovery using machine learning. In *ACM CODASPY*. 85–96.

[34] L. Guan, J. Lin, B. Luo, J. Jing, and J. Wang. 2015. Protecting Private Keys Against Memory Disclosure Attacks Using Hardware Transactional Memory. In *IEEE Symposium on Security and Privacy*. 3–19.

[35] Y. Han, W. Lu, and S. Xu. 2014. Characterizing the Power of Moving Target Defense via Cyber Epidemic Dynamics. In *HotSoS*. 1–12.

[36] Yujuan Han, Wenlian Lu, and Shouhuai Xu. 2020. Preventive and Reactive Cyber Defense Dynamics with Ergodic Time-dependent Parameters Is Globally Attractive. *CoRR* abs/2001.07958 (2020).

[37] K. Harrison and S. Xu. 2007. Protecting Cryptographic Keys from Memory Disclosures. In *IEEE/IFIP DSN'07*. 137–143.

[38] C. Herley and P. C. v. Oorschot. 2017. SoK: Science, Security and the Elusive Goal of Security as a Scientific Pursuit. In *2017 IEEE Symposium on Security and Privacy (SP)*. 99–120.

[39] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. 1997. Proactive public key and signature schemes. In *Proc. ACM CCS*. 100–110.

[40] H. Hethcote. 2000. The Mathematics of Infectious Diseases. *SIAM Rev.* 42, 4 (2000), 599–653.

[41] J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. Rajagopalan, and A. Singhal. 2013. Aggregating vulnerability metrics in enterprise networks using attack graphs. *J. Comput. Secur.* 21, 4 (2013), 561–597.

[42] K. Hoover. 2010. Idealizing Reduction: The Microfoundations of Macroeconomics. *Erkenntnis* 73 (2010), 329–347. Issue 3.

[43] E. Hutchins, M. Cloppert, and R. Amin. 2011. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. In *International Conference on Information Warfare and Security*.

[44] G. Itkis and L. Reyzin. 2002. SiBIR: Signer-Base Intrusion-Resilient Signatures. In *Crypto'02*. 499–514.

[45] A. Kantchelian, M. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. Joseph, and J. Tygar. 2015. Better Malware Ground Truth: Techniques for Weighting Anti-Virus Vendor Labels. In *Proc. AISec*. 45–56.

[46] J. Kephart and S. White. 1991. Directed-Graph Epidemiological Models of Computer Viruses.. In *IEEE Symposium on Security and Privacy*. 343–361.

[47] J. Kephart and S. White. 1993. Measuring and Modeling Computer Virus Prevalence. In *IEEE Symposium on Security and Privacy*. 2–15.

[48] W. Kermack and A. McKendrick. 1927. A Contribution to the Mathematical Theory of Epidemics. *Proc. of Roy. Soc. Lond. A* 115 (1927), 700–721.

[49] Seulbae Kim, Seunghoon Woo, Heejo Lee, and Hakjoo Oh. 2017. VUDDY: a scalable approach for vulnerable code clone discovery. In *Proceedings of 2017 IEEE Symposium on Security and Privacy, San Jose, CA, USA*. 595–614.

[50] Alexander Kott. 2014. *Towards Fundamental Science of Cyber Security.* Springer New York, New York, NY, 1–13.

[51] Alexander Kott. 2018. Challenges and Characteristics of Intelligent Autonomy for Internet of Battle Things in Highly Adversarial Environments. In *2018 AAAI Spring Symposia*. AAAI Press.

[52] Alexander Kott and Ethan Stump. 2019. Intelligent Autonomous Things on the Battlefield. *CoRR* abs/1902.10086 (2019).

[53] A. Kott and P. Théron. 2020. Doers, Not Watchers: Intelligent Autonomous Agents Are a Path to Cyber Resilience. *IEEE Secur. Priv.* 18, 3 (2020), 62–66.

[54] D. Li, Q. Li, Y. Ye, and S. Xu. 2020. Enhancing Deep Neural Networks Against Adversarial Malware Examples. *CoRR* abs/2004.07919 (2020).

[55] D. Li, Q. Li, Y. Ye, and S. Xu. 2020. SoK: Arms Race in Adversarial Malware Detection. *CoRR* abs/2005.11671 (2020).

[56] X. Li, P. Parker, and S. Xu. 2007. Towards Quantifying the (In)Security of Networked Systems.. In *21st IEEE International Conference on Advanced Information Networking and Applications (AINA'07)*. 420–427.

[57] Xiaohu Li, Paul Parker, and Shouhuai Xu. 2011. A Stochastic Model for Quantitative Security Analyses of Networked Systems. *IEEE Transactions on Dependable and Secure Computing* 8, 1 (2011), 28–43.

[58] Zhen Li, Deqing Zou, Shouhuai Xu, Zhaoxuan Chen, Yawei Zhu, and Hai Jin. 2020. VulDeeLocator: A Deep Learning-based Fine-grained Vulnerability Detector. *CoRR* abs/2001.02350 (2020).

[59] Zhen Li, Deqing Zou, Shouhuai Xu, Hai Jin, Hanchao Qi, and Jie Hu. 2016. VulPecker: an automated vulnerability detection system based on code similarity analysis. In *Proceedings of the 32nd Annual Conference on Computer Security Applications, Los Angeles, CA, USA*. 201–213.

[60] Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu, Z. Chen, S. Wang, and J. Wang. 2018. SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities. *CoRR* abs/1807.06756 (2018).

[61] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, and Y. Zhong. 2018. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection. In *Proc. NDSS'18*.

[62] T. Liggett. 1985. *Interacting Particle Systems.* Springer.

[63] Zongzong Lin, Wenlian Lu, and Shouhuai Xu. 2019. Unified Preventive and Reactive Cyber Defense Dynamics Is Still Globally Convergent. *IEEE/ACM Trans. Netw.* 27, 3 (2019), 1098–1111.

[64] W. Lu, S. Xu, and X. Yi. 2013. Optimizing Active Cyber Defense Dynamics. In *Proc. GameSec'13*. 206–225.

[65] M. Lucia, A. Newcomb, and A. Kott. 2019. Features and Operation of an Autonomous Agent for Cyber Defense. *CoRR* abs/1905.05253 (2019).

[66] T. Malkin, D. Micciancio, and S. Miner. 2002. Efficient Generic Forward-Secure Signatures with an Unbounded Number Of Time Periods. In *Proc. EUROCRYPT 2002 (Lecture Notes in Computer Science)*. 400–417.

[67] A. McKendrick. 1926. Applications of Mathematics to Medical Problems. *Proc. of Edin. Math. Soceity* 14 (1926), 98–130.

[68] J. Mireles, E. Ficke, J. Cho, P. Hurley, and S. Xu. 2019. Metrics Towards Measuring Cyber Agility. *IEEE T-IFS* 14, 12 (2019), 3217–3232.

[69] A. Mohaisen and O. Alrawi. 2014. AV-Meter: An Evaluation of Antivirus Scans and Labels. In *Proc. DIMVA*. 112–131.

[70] Jose Morales, Shouhuai Xu, and Ravi Sandhu. 2012. Analyzing Malware Detection Efficiency with Multiple Anti-Malware Programs. In *Proc. CyberSecurity*.

[71] Y. Moreno, R. Pastor-Satorras, and A. Vespignani. 2002. Epidemic Outbreaks in Complex Heterogeneous Networks. *European Physical Journal B* 26 (2002), 521–529.

[72] S. Neuhaus, T. Zimmermann, C. Holler, and A. Zeller. 2007. Predicting vulnerable software components. In *Proceedings of 2007 ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA*. 529–540.

[73] M. Newman. 2003. The structure and function of complex networks. *SIAM Rev.* 45 (2003), 167.

[74] David Nicol, Bill Sanders, Jonathan Katz, Bill Scherlis, Tudor Dumitra, Laurie Williams, and Munindar P. Singh. [n.d.]. The Science of Security 5 Hard Problems (August 2015). http://cps-vo.org/node/21590.

[75] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. 2004. Model-Based Evaluation: From Dependability to Security. *IEEE Trans. Dependable Sec. Comput.* 1, 1 (2004), 48–65.

[76] Steven Noel and Sushil Jajodia. 2017. *A Suite of Metrics for Network Attack Graph Analytics.* Springer International Publishing, Cham, 141–176.

[77] Cameron Nowzari, Victor M. Preciado, and George J. Pappas. 2016. Analysis and Control of Epidemics: A Survey of Spreading Processes on Complex Networks. *IEEE Control Systems* 36, 1 (2016), 26–46.

[78] U.S. National Institute of Standards and Technology. April 16, 2018. Framework for Improving Critical Infrastructure Cybersecurity (Version 1.1). https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

[79] P. Parker and . Xu. 2009. A Method for Safekeeping Cryptographic Keys from Memory Disclosure Attacks. In *Proc. INTRUST'09*. 39–59.

[80] R. Pastor-Satorras and A. Vespignani. 2001. Epidemic Dynamics and Endemic States in Complex Networks. *Physical Review E* 63 (2001), 066117.

[81] R. Pastor-Satorras and A. Vespignani. 2002. Epidemic Dynamics in Finite Size Scale-free Networks. *Physical Review E* 65 (2002), 035108.

[82] M. Pendleton, R. Garcia-Lebron, J. Cho, and S. Xu. 2016. A Survey on Systems Security Metrics. *ACM Comput. Surv.* 49, 4 (2016), 62:1–62:35.

[83] Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. 2017. Modeling and predicting extreme cyber attack rates via marked point processes. *Journal of Applied Statistics* 44, 14 (2017), 2534–2563.

[84] Chen Peng, Maochao Xu, Shouhuai Xu, and Taizhong Hu. 2018. Modeling multivariate cybersecurity risks. *Journal of Applied Statistics* 0, 0 (2018), 1–23.

[85] A. Ramos, M. Lazar, R. H. Filho, and J. J. P. C. Rodrigues. 2017. Model-Based Quantitative Network Security Metrics: A Survey. *IEEE Communications Surveys Tutorials* 19, 4 (2017), 2704–2734.

[86] Rosana Montanez Rodriguez, Edward Golob, and Shouhuai Xu. 2020. Human Cognition through the Lens of Social Engineering Cyberattacks. *CoRR (to appear in Frontiers in Psychology-Cognition)* abs/2007.04932 (2020).

[87] A. Roque, K.. Bush, and C. Degni. 2016. Security is about control: insights from cybernetics. In *Proc. HotSoS*. 17–24.

[88] Fred Schneider. 2011. *Blueprint for a Science of Cybersecurity.* Technical Report. Cornell University.

[89] National Science and Technology Council. 2011. Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program. https://www.nitrd.gov/SUBCOMMITTEE/csia/Fed_Cybersecurity_RD_Strategic_Plan_2011.pdf.

[90] Claude E. Shannon. 1949. Communication theory of secrecy systems. *Bell Syst. Tech. J.* 28, 4 (1949), 656–715.

[91] J. Spring, T. Moore, and D. Pym. 2017. Practicing a Science of Security: A Philosophy of Science Perspective. In *Proc. NSPW*. 1–18.

[92] Salvatore J. Stolfo, Steven M. Bellovin, and David Evans. 2011. Measuring Security. *IEEE Security & Privacy* 9, 3 (2011), 60–65.

[93] Blake Strom. 2018. ATT&CK 101: Cyber Threat Intelligence. https://www.mitre.org/capabilities/cybersecurity/overview/cybersecurity-blog/attck-101

[94] P. Théron and A. Kott. 2019. When Autonomous Intelligent Goodware Will Fight Autonomous Intelligent Malware: A Possible Future of Cyber Defense. In *IEEE MILCOM*. 1–7.

[95] P. Théron, A. Kott, M. Drasar, K. Rzadca, B. Leblanc, M. Pihelgas, L. Mancini, and F. Gaspari. 2020. Reference Architecture of an Autonomous Agent for Cyber Defense of Complex Military Systems. In *Adaptive Autonomous Secure Cyber Systems*. Springer, 1–21.

[96] Piet Van Mieghem, Jasmina Omic, and Robert Kooij. 2009. Virus spread in networks. *IEEE/ACM Trans. Netw.* 17, 1 (Feb. 2009), 1–14.

[97] L. Wang, S. Jajodia, and A. Singhal. 2017. *Network Security Metrics.* Springer.

[98] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel. 2014. k-Zero Day Safety: A Network Security Metric for Measuring the Risk of Unknown Vulnerabilities. *IEEE TDSC* 11, 1 (2014), 30–44.

[99] Q. Wang, W. Guo, K. Zhang, A. Ororbia II, X. Xing, X. Liu, and C. Giles. 2017. Adversary Resistant Deep Neural Networks with an Application to Malware Detection. In *ACM KDD*. 1145–1153.

[100] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. 2003. Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint. In *IEEE SRDS'03*. 25–34.

[101] L. Xu, L. Chen, Z. Gao, X. Fan, K. Doan, S. Xu, and W. Shi. 2019. KCRS: A Blockchain-Based Key Compromise Resilient Signature System. In *International Conference on Blockchain and Trustworthy Systems (BlockSys)*. 226–239.

[102] L. Xu, Z. Zhan, S. Xu, and K. Ye. 2013. Cross-layer detection of malicious websites. In *ACM CODASPY*. 141–152.

[103] L. Xu, Z. Zhan, S. Xu, and K. Ye. 2014. An Evasion and Counter-Evasion Study in Malicious Websites Detection. In *IEEE CNS*. 265–273.

[104] Maochao Xu, Gaofeng Da, and Shouhuai Xu. 2015. Cyber Epidemic Models with Dependences. *Internet Mathematics* 11, 1 (2015), 62–92.

[105] M. Xu, L. Hua, and S. Xu. 2017. A Vine Copula Model for Predicting the Effectiveness of Cyber Defense Early-Warning. *Technometrics* 59, 4 (2017), 508–520.

[106] M. Xu, K. M. Schweitzer, R. M. Bateman, and S. Xu. 2018. Modeling and Predicting Cyber Hacking Breaches. *IEEE T-IFS* 13, 11 (2018), 2856–2871.

[107] M. Xu and S. Xu. 2012. An Extended Stochastic Model for Quantitative Security Analysis of Networked Systems. *Internet Mathematics* 8, 3 (2012), 288–320.

[108] Shouhuai Xu. 2007. On the security of group communication schemes. *Journal of Computer Security* 15, 1 (2007), 129–169.

[109] Shouhuai Xu. 2008. Collaborative Attack vs. Collaborative Defense. In *Proc. CollaborateCom*. 217–228.

[110] Shouhuai Xu. 2014. Cybersecurity Dynamics. In *Proc. HotSoS'14*. 14:1–14:2.

[111] S. Xu. 2014. Emergent Behavior in Cybersecurity. In *Proc. HotSoS*. 13:1–13:2.

[112] Shouhuai Xu. 2019. Cybersecurity Dynamics: A Foundation for the Science of Cybersecurity. In *Proactive and Dynamic Network Defense*. 1–31.

[113] S. Xu, X. Li, T. Parker, and X. Wang. 2011. Exploiting Trust-Based Social Networks for Distributed Protection of Sensitive Data. *IEEE T-IFS* 6, 1 (2011), 39–52.

[114] Shouhuai Xu, Wenlian Lu, and Hualun Li. 2015. A Stochastic Model of Active Cyber Defense Dynamics. *Internet Mathematics* 11, 1 (2015), 23–61.

[115] S. Xu, W. Lu, and L. Xu. 2012. Push- and pull-based epidemic spreading in networks: Thresholds and deeper insights. *ACM TAAS* 7, 3 (2012).

[116] S. Xu, W. Lu, L. Xu, and Z. Zhan. 2014. Adaptive Epidemic Dynamics in Networks: Thresholds and Control. *ACM TAAS* 8, 4 (2014).

[117] S. Xu, W. Lu, and . Zhan. 2012. A Stochastic Model of Multivirus Dynamics. *IEEE Transactions on Dependable and Secure Computing* 9, 1 (2012), 30–45.

[118] S. Xu and M. Yung. 2009. Expecting the Unexpected: Towards Robust Credential Infrastructure. In *Financial Crypto*. 201–221.

[119] W. Xu, Y. Qi, and D. Evans. 2016. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers. In *NDSS*.

[120] F. Yamaguchi, M. Lottmann, and K. Rieck. 2012. Generalized vulnerability extrapolation using abstract syntax trees. In *ACSAC*. 359–368.

[121] F. Yamaguchi, C. Wressnegger, H. Gascon, and K. Rieck. 2013. Chucky: exposing missing checks in source code for vulnerability discovery. In *ACM CCS*. 499–510.

[122] L. Yang, P. Li, X. Yang, and Y. Tang. 2018. A risk management approach to defending against the advanced persistent threat. *IEEE TDSC* (2018), 1–1.

[123] L. Yang, X. Yang, and Y. Tang. 2018. A Bi-Virus Competing Spreading Model with Generic Infection Rates. *IEEE Trans. Netw. Sci. Eng.* 5, 1 (2018), 2–13.

[124] Z. Zhan, M. Xu, and S. Xu. 2013. Characterizing Honeypot-Captured Cyber Attacks: Statistical Framework and Case Study. *IEEE T-IFS* 8, 11 (2013).

[125] Z. Zhan, M. Xu, and S. Xu. 2014. A Characterization of Cybersecurity Posture from Network Telescope Data. In *Proc. InTrust*. 105–126.

[126] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu. 2015. Predicting Cyber Attack Rates With Extreme Values. *IEEE Transactions on Information Forensics and Security* 10, 8 (2015), 1666–1677.

[127] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese. 2016. Network Diversity: A Security Metric for Evaluating the Resilience of Networks Against Zero-Day Attacks. *IEEE Trans. Inf. Forensics Secur.* 11, 5 (2016), 1071–1086.

[128] R. Zheng, W. Lu, and S. Xu. 2015. Active Cyber Defense Dynamics Exhibiting Rich Phenomena. In *Proc. HotSoS*.

[129] R. Zheng, W. Lu, and S. Xu. 2018. Preventive and Reactive Cyber Defense Dynamics Is Globally Stable. *IEEE TNSE* 5, 2 (2018), 156–170.

[130] D. Zou, S. Wang, S. Xu, Z. Li, and H. Jin. 2020. μVulDeePecker: A Deep Learning-Based System for Multiclass Vulnerability Detection. *IEEE TDSC* (2020).