IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 28, 2021, accepted May 26, 2021, date of publication June 8, 2021, date of current version June 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3087467

Simulating Robustness of Structural Controllability for Directed Networks Under Multi-Round Edge Strategies

B. ALWASEL
Department of Applied Natural Science, College of Unaizah Community, Qassim University, Unaizah 51911, Saudi Arabia
e-mail: bwasel@qu.edu.sa

This work was supported by the Deanship of Scientific Research, Qassim University.

ABSTRACT The study of structural controllability of control systems is a crucial property in the design and analysis of complex networks as well as networks which require a control relationship between nodes. The fundamental aim of attack vulnerability research is to safeguard electric power networks along-with their control systems as part of critical infrastructure systems. Such a system may have its structural control undermined or co-opted to hinder or hijack control if the entire network system is already known and understood by an attacker. A significant focus on the graph-theoretical interpretation of Kalman controllability has emerged as a concept linked to structural controllability that offers a powerful abstraction for understanding the structural properties of a control network and its critical elements. The determination of driver node sets that can monitor the whole network is therefore enabled, although it is a $W[2]$-hard problem identifying these nodes. Indeed, problematic computational complexity is a feature of the various extant driver node identification techniques. Accordingly, this paper is highly motivated to adopt the power dominating set approach to explore how directed Erdős-Rényi networks are influenced by targeted iterative multiple-edge removal, in addition to the assessment of its effects on the robustness of network controllability from multiple structural vulnerabilities.

INDEX TERMS Structural controllability, network robustness, attack models, cyber-physical systems.

I. INTRODUCTION

The studies on securing networked control systems located within natural, economical and man-made engineered systems have attracted many researchers from both fields of network science and control science [1]. In the viewpoint of complex networks, individuals comprise the nodes, whilst the connections they share act as the edges. Driver nodes in electric power networks, for instance, can constitute control terminal units that guide industrial sensors or actuators. Malicious attacks can remove edges, which can lead to the violation of real-time boundaries. Thus, the redistribution loads across the whole network can enlarge the load of some other edges, which may be more than they can handle. The network control can be deteriorated as its observability experiences substantial reduction. Consequently, a range-based attack on edges represents a significant concern in control systems [2]–[4]; if such attacks are not guarded against, the attacker can create more disruptions. This attack scenario could leave two states of the network unable to connect in a time-dependent input. As a result, the control robustness of a network in safeguarding against the failure of any integral components is a significant issue in relation to the operation of a complex network [1]. This issue has become a further considerable problem in network controllability and its robustness, which has been broadly studied, in particular following the examination put forward by Lin [5] on structural controllability.

To design and maintain a networked system under control, two structural properties of the dynamical systems have been well established as observability and controllability. However, the focus on substantial complex systems and networks as the environment for these concepts has renewed the researchers' interest recently [6]–[8].

Kalman [9] initially considered state controllability and observability as properties for linear time-invariant (LTI) systems. Informally, controllability is defined as the ability to derive the requisite configuration from an arbitrary configuration in a finite number of steps. Linear network models are the specific initiation point in the study of network

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

controllability. Therefore, the focus of this paper is a linear time-invariant system, with taking into consideration the following equation representing this system:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t), \quad x(t_0) = x_0 \tag{1}$$

where $x(t) \in R^n$ is the state vector at time t, $u(t) \in R^m$ is the input vector through which network dynamics may be influenced; $\mathbf{A}$ is the state matrix of the system's representative network, while the interacting components are indicated by every non-zero input. $\mathbf{B} \in R^{n\times m}$ is the input matrix ($m \leq n$) stipulates the set of nodes controlled by a time-dependent input vector $u(t) = (u_1(t), \ldots, u_m(t))$, with the requisite state forced by this. The system in equation (1) is controllable if and only if:

$$\text{rank}\,[\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{n-1}\mathbf{B}] = n \tag{2}$$

To determine whether an LTI system is controllable or observable, one could verify the rank of the constant controllability or observability matrix of the system, also known as the Kalman rank condition for controllability or observability [9]. However, the inappropriateness of the Kalman rank condition is apparent from identifying precise system parameters in an applied context. Consequently, a graph perspective concerning controllability analysis was offered through Lin's notion of structural controllability, which considers network or system parameters and can resolve this challenge [5]. The seminal work by Liu *et al*. proposed that a bipartite graph used for conversion of the structural controllability problem into a maximum matching problem [10]; this also helps to identify the necessary minimum number of driver nodes (N_D) or the minimum number of inputs required to control a network by using a minimum inputs theorem. Before stating the relevant theorem, some fundamental definitions are required to describe the network structure characteristics:

Definition 1 (Stem and Bud, [5]): Given a directed graph $G(\mathbf{A}, \mathbf{B}) = (V_\mathbf{A} \cup V_\mathbf{B}, E_\mathbf{A} \cup E_\mathbf{B})$, a stem is a directed path originating from any node of $V_\mathbf{B}$, while a bud is a directed cycle with an additional edge that ends, but does not begin, in a vertex of the cycle; this edge is known as the distinguished edge.

Definition 2 (Dilation, [5]): Given a digraph $G(\mathbf{A}, \mathbf{B}) = (V_\mathbf{A} \cup V_\mathbf{B}, E_\mathbf{A} \cup E_\mathbf{B})$, $G(\mathbf{A}, \mathbf{B})$ contains a dilation if and only if there is a subset $S \subset V_\mathbf{A}$ such that $|S| > |T(S)|$, where $T(S)$ is the neighbourhood set of a set S representing the tails of edges whose heads are all vertices of S.

Definition 3 (Inaccessibility, [5]): Given a digraph $G(\mathbf{A}, \mathbf{B}) = (V_\mathbf{A} \cup V_\mathbf{B}, E_\mathbf{A} \cup E_\mathbf{B})$ and a state node v_i of $V_\mathbf{A}$, node v_i is inaccessible, if and only if there are no directed paths that reach v_i from the input vertices of $V_\mathbf{B}$.

Definition 4 (Cactus, [5]): A cactus is a subgraph that can be defined recursively as follows: A stem is a form of cactus, thus, given a stem S_0 and buds $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_l$, then $S_0 \cup \mathbf{B}_1 \cup \mathbf{B}_2 \cup \ldots \cup \mathbf{B}_l$ is a cactus if for every i ($1 \leq i \leq l$) the initial vertex of the distinguished edge of $\mathbf{B}_i$ is not the top of S_0 and it is the only vertex that belongs simultaneously to $\mathbf{B}_i$ and $S_0 \cup \mathbf{B} \cup \mathbf{B}_2 \cup \ldots \cup \mathbf{B}_{i-1}$.

Theorem 1 (Lin's Structural Controllability Theorem, [5]): Given system $(\mathbf{A}, \mathbf{B})$ described by equation (1) is said to be structurally controllable if a linear control system $(\mathbf{A}, \mathbf{B})$ is structurally controllable, where a directed graph $G(\mathbf{A}, \mathbf{B})$ does not include any inaccessible node or dilation such that the $G(\mathbf{A}, \mathbf{B})$ is spanned by a cactus.

Nevertheless, this paper concentrates on the similar power dominating set (PDS) problem, which Haynes *et al.* [11] developed to build on the PDS. The principal reason for this is the structure of electric power networks, and these networks requiring the provision of efficacious control. Adopting the PDS problem or the maximum matching problem for bipartite digraphs is the requisite initial stage so that identification of the minimal set of nodes $V_\mathbf{B}$ in $G(\mathbf{A}, \mathbf{B}) = (V, E)$ from a given $G(\mathbf{A}, \mathbf{B}) = (V, E)$ is possible, as well as to use observed nodes $V_\mathbf{A}$ and driver nodes $V_\mathbf{B}$ to convey a graphical design [10]. Each of the problems undertakes a node-by-node analysis of the whole graph, in addition to assessing the degree of dominance for the nodes in relation to their neighbourhood. The observed nodes, denoted as O, and the minimum subset of driver nodes (N_D) are two crucial sets that can be derived from this process, with a minimum of one driver node involved in their control $O \leftarrow V \setminus N_D$.

The contribution of this paper is, therefore, to investigate the behaviour of network controllability in directed Erdős-Rényi (ER) networks when subject to multi-round edge removal in various scenarios using the power dominating set problem. The robustness of structural controllability over a directed ER network and its observability before and after an attack is then assessed by simulating the attack scenarios proposed here. In terms of practicality, the findings shown in this paper are significant. They can be applied to evaluate vulnerability analysis on edge attacks (*e.g.* transmission lines or communication links joining two electrical sensors or actuators in remote monitoring real-world systems such as electrical power network control). The restoration strategies under perturbations are not the focus of this paper.

The remaining sections of the paper are structured as follows. Section II gives a brief review of the relationship between structural controllability and power dominance including a number of recent studies on the attack vulnerability of network controllability. Section III describes the network model underpinned by diverse types of multi-round edge attack strategies on robustness. Subsequently, Section IV details how such disturbance strategies impact network controllability and observability, discussing the quantitative analysis and findings of the network controllability under vulnerability for directed ER networks. Finally, Section V concludes the paper.

II. STRUCTURAL CONTROLLABILITY AND POWER DOMINATION

Equation (2) shows the controllability rank condition, which provides a thorough framework for the design and analysis of control systems. Thus, the computation of this criterion in an arbitrary network requires knowing the weight of each link, that is either not known for many real networks or is time-dependent and approximated. Nevertheless, should the weights be made clear, a brute-force search is still needed to calculate Kalman's rank criterion for $(2^N - 1)$ clear-cut combinations that can prove costly for large complex networks.

Given a system described by equation (1), the matrix $\mathbf{A}$ indicates the network topology, while the matrix $\mathbf{B}$ is the input matrix, which shows the nodes where the external controllers are injected into the entire network. These nodes are also referred to as driver nodes (N_D) and correspond to the input vector u. Lin [5] showed that the whole system, denoted as $(\mathbf{A}, \mathbf{B})$, can be illustrated by a directed graph $G(\mathbf{A}, \mathbf{B}) = (V, E)$ with $V = V_\mathbf{A} \cup V_\mathbf{B}$ is the set of vertices and $E = E_\mathbf{A} \cup E_\mathbf{B}$ is the set of edges.

Acquiring the minimal set of $V_\mathbf{B}$ (driver nodes) from a provided $G(V, E)$ requires the application of the PDS problem or the maximum matching problem for bipartite digraphs. Even though the legitimacy of the maximum matching method for extracting N_D has been evidenced through other studies [1], [10], [12]–[14], the PDS problem is the concentration of this paper. The PDS problem was originally suggested by Haynes *et al.* [11] to study electric power networks and the expansion of the well-known Dominating Set (DS) problem. Ultimately, Haynes *et al.* first devised **OR1** and **OR2** as the key observation rules, subsequently simplified by Kneis *et al.* [15], which primarily support the extraction of N_D through the PDS:

[OR1] A vertex in N_D observes itself and all of its neighbours.

[OR2] If an observed vertex v of degree $d^+ \geq 2$ is adjacent to $d - 1$ observed vertices, then the remaining unobserved neighbour becomes observed as well.

It is possible to deduce from this definition that **OR1** is included within the definition of **OR2**, implying that the subset of nodes that conform to **OR1** is also part of the subset of nodes conforming to **OR2**. Therefore, compliance with each of the rules is necessary for control, while any topological change may indicate an error in **OR1-2** compliance and, subsequently, the system's deterioration. Furthermore and notably, application of the two rules to the dual problem of controllability is being undertaken here, despite their characterisation as observation rules. Also, it could be noted that the only distinction between PDS and DS problems is the presence of **OR2**, and DS is proven to be
textnp-complete for general graphs with a polynomial-time approximation factor of $\Theta(\log n)$ [16]. The PDS problem, on the other hand, is a generalization of the DS problem, and Haynes *et al.* have shown that it is still
textnp-complete for general graphs and valid for certain specific types of graphs such as bipartite graphs and chordal graphs [11], [17]. Similarly, a power dominating set with the minimum cardinality of a given digraph is also **NP**-complete, as shown by Aazami and Stilp [18] and cannot be approximated better than $\mathbf{NP} \subseteq DTIME(n^{polylog(n)})$.

A. CONTROLLABILITY OF NETWORKS UNDER VULNERABILITY

When the network distribution and its power domination are exposed to vulnerability attack, an adversary may disrupt a distributed system or prevent defenders from recovering full or partial control of the network; this provides a powerful incentive to analyse vulnerabilities on robustness controllability when network edges are susceptible to malicious attacks or random failures. Various recent studies on complex networks subjected to malicious attacks and random failures have sought to measure the attack vulnerability of numerous complex network systems, such as real-world networks where removal of some of the edges or nodes has occurred [19]–[21]. Pu *et al.* investigated how cascading failures and attacks impacted directed Erdős-Rényi and scale-free networks in relation to network controllability [1]. Cascading overload failures as a result of the removal of vertices because of random or intentional attacks was assessed by the researchers in [22]; a network part or utter collapse can result. The robustness of network controllability on a number of network topologies in the presence of vertex removal was investigated by [23], as well as the effect of several non-interactive attack types on the PDS and underlying graphs. The researchers also considered range-based attacks on edges, which are interesting because edges have been overlooked through the emphasis on attacks on nodes in most complex network security research.

Additionally, a dynamic programming algorithm based on recent work by Aazami and Stilp as well as Guo *et al.* [17], [18] was designed to compute PDS in the context of structural controllability recovery after a malicious attack on network vertices [24]. This approach is based on a nice tree decomposition for a given ER random digraph in a LTI model, where the worst-case time complexity is $O(nc^k)$, and average-case time complexity is $O(\log(c^k))$. As a result, we proposed a novel power dominating set algorithm that recovers a control network by re-using the remaining PDS of the original where possible [25]. This approach based on depth-first search yields an improved average-case complexity over previous work in [24], while the worst-case time complexity remains unchanged. Following that, using a block decomposition on the input digraph, a restoration method for reconstructing a minimal PDS, when the PDS or its dependent nodes partially compromised, was studied [26]. Besides, Alcaraz *et al.* proposed three strategies to efficiently restore structural controllability of general power-law and scale-free digraphs following attacks [27]. The authors of [28] studied the ability to recovers the minimum-input structural controllability of digraph in linear time by identifying a maximum matching without recomputation. They also devised an approach to efficiently recovering structural controllability of the residual system following malicious attacks or failures by introducing a minimum set of edges into a given system network [29], as well as the classification of the effects of removing single node driver on controlling residual network [30].

III. NETWORK AND ATTACK MODELS

This section covers the graph class as well as several attack strategies. The network model is built on a directed ER random graph since it is one of the oldest and most well-studied network models, and is widely used to model a range of complex networks, allowing for the analysis of various network processes such as cascading failures.

A. NETWORK MODEL

To examine the robustness of controllability for directed ER networks under vulnerability, the random directed graphs $G(V, E)$ are studied, provided by Erdős-Rényi random graph class $ER(n, p)$ which is defined as follows [31]:

Definition 5 (Erdős-Rényi Random Graph): The $ER(n, p)$ model has two boundaries, n and p. Here n is the number of vertices of the graph and p is the edge probability. The random connection of nodes allows for the construction of a graph. The edges featured in graph G are determined independently with the edge probability p so that the pairs of vertices $u, v \in n$ connect with an identical edge probability. Equally, the graphs with n nodes and M edges have the same $p^M(1-p)^{\binom{n}{2}-M}$) probability.

For the network model, it is assumed that a given input network G has an arbitrary set of nodes V and a set of edges E, with no self-loops or duplicate edges (i.e. two edges with both the same tail vertex and the same head vertex). Subsequently, networks with small (≥ 100) and large (≤ 2000) numbers of nodes are modelled, in which any two nodes are adjacent with independent probability p for each node pair. The resulting instance of $ER(n, p)$ is a weakly connected graph, where the underlying undirected graph is connected and without its isolated vertices (i.e. a vertex with in-degree and out-degree zero, denoted here as $V_{isolated}$).

TABLE 1. The simulation results of the computation of PDS (or set of N_D) for several directed ER network sizes with different small connectivity probabilities.

N	p	E	PDS	$N_{connected}$	$V_{isolated}$
100	0.031	153	23	72	5
500	0.0050	624	142	313	45
1000	0.0025	1249	294	615	91
2000	0.0012	2399	582	1233	185

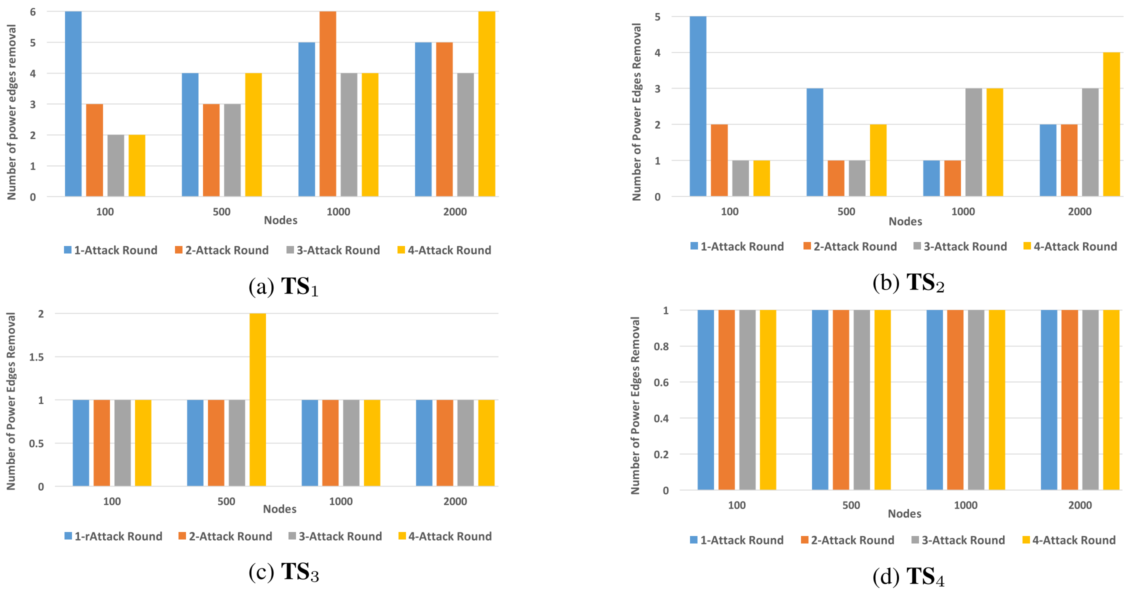
Since several real-world networks such as real power networks are sparse, different network sizes were generated by an ER model with 100, 500, 1000 and 2000 nodes and with several low connectivity probabilities. Based on the previous work [32], the number of PDS (a set of driver nodes) for the directed ER networks presented here was computed, as shown in Table 1, as well as the result of the graphical representation of network controllability for a network of 500 nodes as an example (see Figure 2). This algorithm used the structural controllability abstraction, which offers an equivalent formulation for identifying minimum driver node subsets. It relied on the PDS formulation to traverse the entire network to search for the best driver candidates N_D (i.e. PDS) that met the **OR1** and **OR2** conditions, as shown in Pseudocode 1. These obtained driver nodes are not unique and are achievable by applying the two observation rules for controllability as shown in the two observation rules **OR1** and **OR2** above, where **OR1** involves N_D controlling all vertices in $V \setminus N_D$ by the application of **OR2**. The computational findings in Table 1 illustrate that as the number of nodes in the original networks increases, the minimum number of PDS increases as well, owing to the networks' low connectivity probabilities, which reduce the total number of edges in the networks.

TABLE 2. Network connectivity (C) and control diameter (D) before and after further rounds of attacks.

Threat Scenarios		N							
		100		500		1000		2000	
		C	D	C	D	C	D	C	D
TS_1	Before Attack	6	24	4	9	5	17	5	6
	1-AR	0	0	0	0	0	0	0	0
	Before Attack	3	2	3	7	6	42	5	5
	2-AR	0	0	0	0	0	0	0	0
	Before Attack	2	3	3	4	4	5	4	5
	3-AR	0	0	0	0	0	0	0	0
	Before Attack	2	3	4	9	4	6	6	33
	4-AR	0	0	0	0	0	0	0	0
TS_2	Before Attack	6	24	4	9	5	17	5	6
	1-AR	1	5	1	9	4	17	3	6
	Before Attack	3	2	3	7	6	42	5	5
	2-AR	1	1	2	5	5	42	3	4
	Before Attack	2	3	3	4	5	42	4	5
	3-AR	1	3	2	4	2	41	1	5
	Before Attack	2	3	4	9	4	17	6	33
	4-AR	1	3	2	6	1	13	2	17
TS_3	Before Attack	2	2	1	1	1	1	2	6
	1-AR	1	1	0	0	0	0	1	6
	Before Attack	1	1	2	2	1	1	2	3
	2-AR	0	0	1	1	0	0	1	3
	Before Attack	1	1	1	1	2	2	1	5
	3-AR	0	0	0	0	1	2	0	0
	Before Attack	2	3	5	33	2	1	1	1
	4-AR	1	3	3	4	1	1	0	0
TS_4	Before Attack	2	24	2	9	2	7	2	44
	1-AR	1	8	1	9	1	4	1	44
	Before Attack	2	8	2	33	2	42	2	5
	2-AR	1	8	1	33	1	42	1	4
	Before Attack	2	8	2	4	2	17	2	6
	3-AR	1	8	1	2	1	17	1	6
	Before Attack	2	3	2	4	2	3	2	3
	4-AR	1	3	1	1	1	1	1	3

B. ATTACK STRATEGIES

So as to analyse the vulnerability of controllability under directed ER networks in terms of network connectivity and observability (as the dual of controllability), the paper investigates the behaviour of network controllability when a network is exposed to a range of edge attacks that might damage



(a) $\mathbf{TS}_1$ (b) $\mathbf{TS}_2$ (c) $\mathbf{TS}_3$ (d) $\mathbf{TS}_4$

FIGURE 1. The implications of eliminating a driver node or its dependent are identified by calculating the number of removed edges when are exposed to multiple-round attacks ($\mathbf{TS}_1$, $\mathbf{TS}_2$, $\mathbf{TS}_3$ and $\mathbf{TS}_4$).

a control network by eliminating its existing driver nodes or isolating the network completely or partially by deleting all or some edges from the network. Here, it can be supposed that the attackers are familiar with the structural control of the deployed networks and exploit existing vulnerabilities to execute malicious removals of edges from nodes in the current N_D or dependent nodes. The following threat scenarios (denoted here as $\mathbf{TS}_i$) are based on the above mentioned type of attack:

$\mathbf{TS}_1$**:** An adversary targets a node in N_D with the largest out-degree (i.e. the number of outgoing edges linking to the most connected nodes) by iterative removal of all its edges.

$\mathbf{TS}_2$**:** Repeatedly attacks structural controllability by deleting a few (but not all) edges from a vertex in N_D with maximum out-degree.

$\mathbf{TS}_3$**:** Randomly deletes some (but not all) edges from a vertex within N_D of the minimum out-degree.

$\mathbf{TS}_4$**:** Continuously removes one edge at most from vertices not within N_D in each attack round.

IV. DISCUSSION

This section analyses the vulnerability of structural controllability for directed ER networks with respect to the attack scenarios defined in Section III through Matlab simulations.[1] Under multi-round edge removal attacks, robustness and vulnerability are assessed from two perspectives:

1) degree of structural connectivity, and
2) degree of structural observability.

For structural connectivity, edge connectivity of driver nodes and their control diameter are considered in addition to disconnected components. For observability, the remaining observable nodes after an attack as a percentage (**OR1**) are computed. For the former perspective, the paper introduces certain connectivity metrics in the context of structural controllability:

Definition 6 (Edge Connectivity): Edge connectivity, denoted by (**C**), is the minimum number of directed out-edges needed to disconnect the dependent nodes from a node within driver nodes (i.e. PDS).

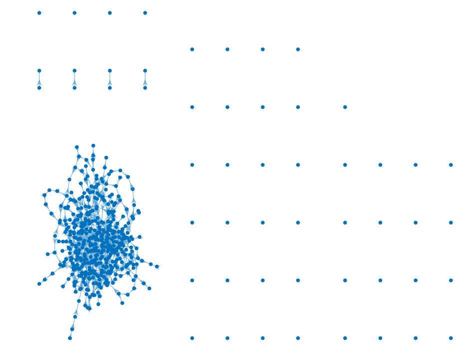
Definition 7 (Control Diameter): Control diameter, denoted by (**D**), is defined as the greatest length of the shortest dependency path between a node in driver nodes (i.e. PDS) and its dependent nodes, such that the edges of the path are directed from a node within PDS to a leaf (child) node.

Definition 8 (Disconnected Component): Let $u \in PDS$, a node v is said to be controlled if there is a directed path from u to v. Each directed path that is incident to u is a dependency path. Dependency paths can be defined as paths where a sequence of nodes is directed from u to v as a connected component. Therefore, the deletion of edges in a dependency path results in the emergence of new disconnected components, denoted by (**DCC**), see Figure 6.

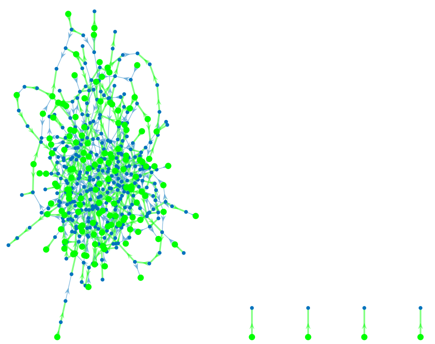
A. EXPERIMENTAL RESULTS

The simulation is carried out as follows. It is assumed that adversaries with pre-existing knowledge of structural control of the deployed networks exploit existing vulnerabilities to perform malicious removals of edges from nodes in the current driver nodes N_D or their dependent nodes. Here four attack rounds (referred to as **i-AR**) are applied based on different threat scenarios ($\mathbf{TS}_1$, $\mathbf{TS}_2$, $\mathbf{TS}_3$ and $\mathbf{TS}_4$) as specified in subsection III-B. The findings of the threat model against various directed ER network sizes are also depicted graphically in Figures 3-6 (see APPENDIX).

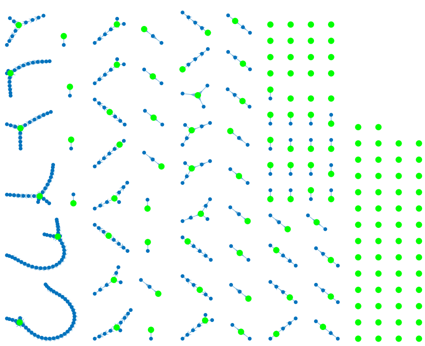
[1]The full code is available in APPENDIX.



(a) An original network of 500 nodes, including its isolated vertices and $p = 0.0050$ before an attack.



(b) The structurally controlled network after the computation of driver nodes.



(c) The representation of each driver node N_D with its dependent nodes. The network is dominated by a minimum set of driver nodes, which are marked in green, while the dependent nodes controlled by N_D are marked in blue, with a dependence path consisting of a sequence of dependent nodes.

FIGURE 2. Illustrations of controlling network.

Consequently, the results show that $\mathbf{TS}_1$ attacks are more efficient on network structural controllability than the other threat scenarios. This result is evident in Table 2, where edge connectedness is completely destroyed because a node in N_D with the highest out-degree is targeted by iterative removal of all its edges. As seen in Table 3, this attack results in a significant reduction in observability, and therefore, the appearance of new disconnected components, in which the affected nodes (denoted as **AN**) are isolated from a network, as shown in Table 4. Furthermore, the degradation of

TABLE 3. Observation rates after perturbations or attacks.

Threat Scenarios	i-AR	N: 100	500	1000	2000
TS_1	1-AR	0.53	0.95	0.95	0.99
	2-AR	0.48	0.91	0.81	0.98
	3-AR	0.41	0.89	0.80	0.97
	4-AR	0.35	0.85	0.78	0.93
TS_2	1-AR	0.60	0.97	0.99	0.99
	2-AR	0.56	0.96	0.99	0.99
	3-AR	0.53	0.95	0.90	0.98
	4-AR	0.52	0.93	0.87	0.96
TS_3	1-AR	0.97	0.99	0.99	0.99
	2-AR	0.95	0.99	0.99	0.99
	3-AR	0.93	0.98	0.99	0.99
	4-AR	0.92	0.91	0.99	0.99
TS_4	1-AR	0.75	0.99	0.99	0.98
	2-AR	0.74	0.98	0.98	0.98
	3-AR	0.72	0.97	0.98	0.98
	4-AR	0.70	0.97	0.98	0.98

TABLE 4. The number of affected nodes (AN) along with disconnected components (DCC) per attack round in each threat scenario.

Threat Scenarios		N							
		100		**500**		**1000**		**2000**	
	i-AR	**AN**	**DCC**	**AN**	**DCC**	**AN**	**DCC**	**AN**	**DCC**
TS_1	1-AR	44	7	22	5	46	6	19	6
	2-AR	5	4	16	4	128	7	15	6
	3-AR	7	3	11	4	12	5	16	5
	4-AR	5	3	18	5	13	5	73	7
TS_2	1-AR	38	5	12	3	4	1	9	2
	2-AR	3	2	7	1	1	1	6	2
	3-AR	3	1	2	1	84	3	10	3
	4-AR	1	1	10	2	28	3	38	4
TS_3	1-AR	2	1	2	1	2	1	2	2
	2-AR	2	1	2	1	2	1	2	2
	3-AR	2	1	2	1	1	1	6	2
	4-AR	1	1	34	2	1	1	2	2
TS_4	1-AR	23	1	6	1	3	1	29	1
	2-AR	1	1	2	1	8	1	3	1
	3-AR	2	1	2	1	4	1	3	1
	4-AR	2	1	3	1	2	1	1	1

network controllability can lead to the entire network malfunctioning if targeted repeatedly by $\mathbf{TS}_1$. In the worst case, if this attack pattern is repeatedly executed until all nodes in the set of driver nodes and all dependent nodes are eliminated, full destruction of the control network can result. While the results confirm that $\mathbf{TS}_2$ can also harm the networks' connectivity, the damage it causes is not as severe as that caused by $\mathbf{TS}_1$, as shown in Table 2. Nonetheless, the networks attacked by $\mathbf{TS}_2$ become very sensitive in connectivity terms, and the impact of compromised nodes is noticeable in both small and large networks when the number of attacks reaches the node connectivity with the highest out-degree. However, there is no remarkable change in the connectivity of the networks when a $\mathbf{TS}_3$ attack occurs, as the behaviour of this attack eliminates some (but not all) edges from a vertex within N_D of the minimum out-degree.

The results obtained also highlight that observability rates dramatically decrease in small networks when subject to

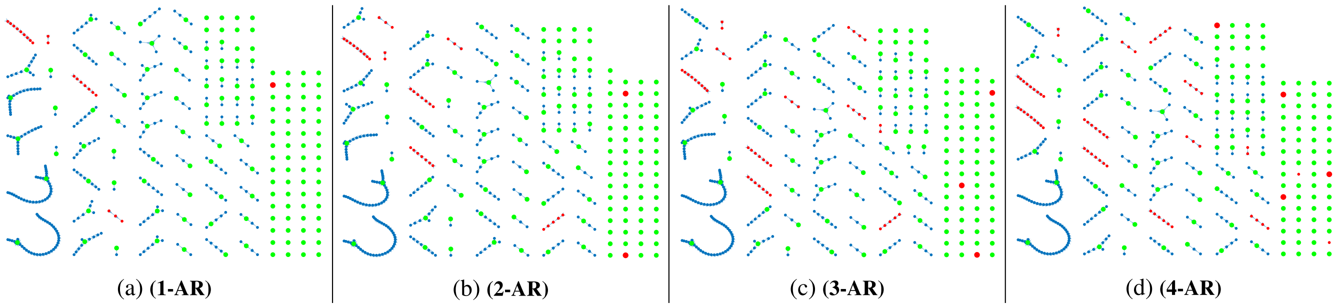


FIGURE 3. The simulation process of network controllability robustness under multiple-round attacks (i-AR). Here, the vulnerability scenario $\mathbf{TS_1}$ is applied to a directed ER network of 500 nodes, in which an adversary targets a node in N_D with the largest out-degree (i.e. the number of outgoing edges linking to the most connected nodes) by iterative removal of all its edges. Nodes with (small) red denote uncontrolled nodes which are completely isolated from a network after an attack, while an attacked driver node (N_D) with all of its edges removed is denoted by (big) red node. A vertex in N_D that is susceptible to the elimination of a few (but not all) edges is represented by an orange node.

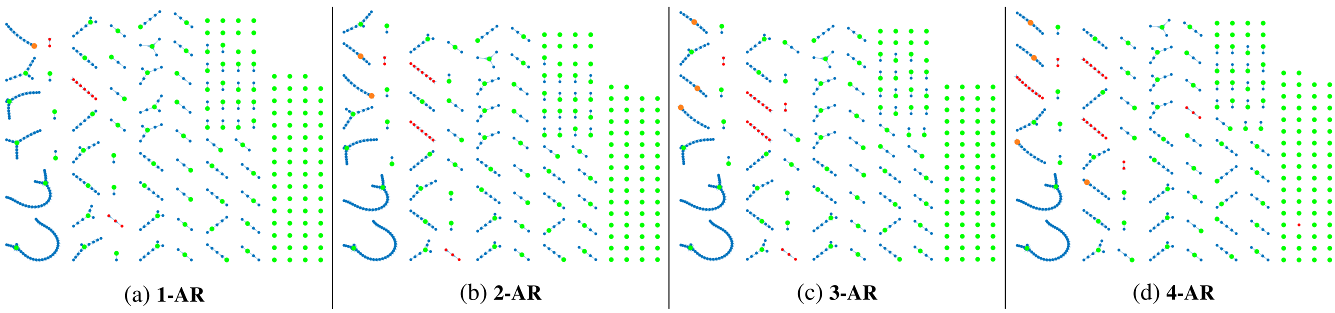


FIGURE 4. The attack strategy $\mathbf{TS_2}$ with four rounds of attack is illustrated here, in which a vertex in N_D with the highest out-degree is repeatedly attacked by removing a few (but not all) of its edges.

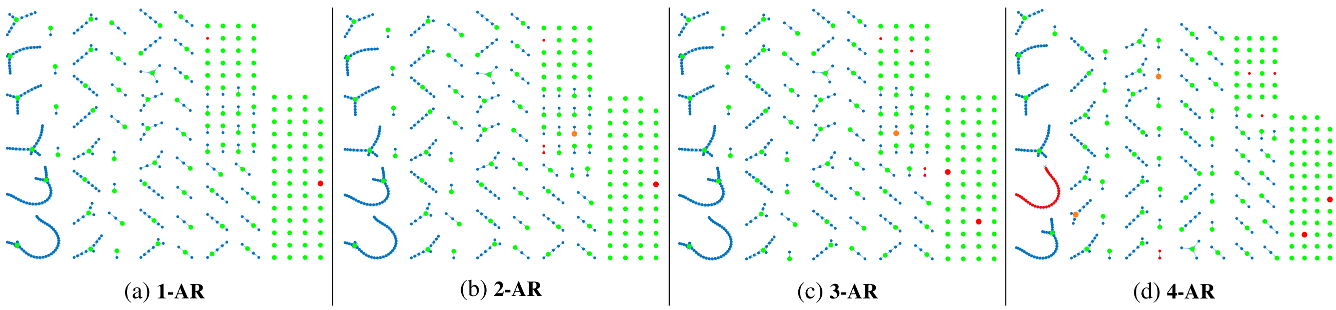


FIGURE 5. Structural controllability of a directed network of 500 nodes is vulnerable to edge deletions based on $\mathbf{TS_3}$, in which some (but not all) edges from a vertex within N_D of the minimum out-degree are randomly removed by four attack rounds.

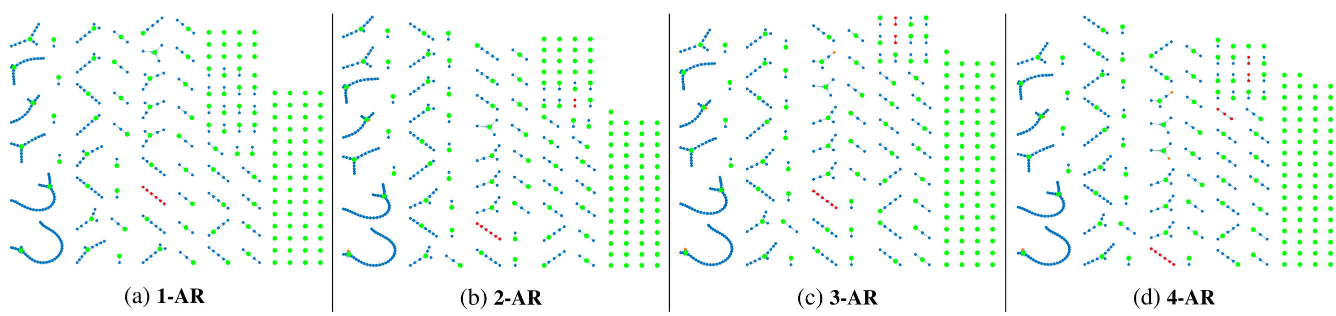


FIGURE 6. This threat $\mathbf{TS_4}$ eliminates one edge at most from vertices not belonging to N_D in each attack round (i-AR). Following an attack, the emergence of disconnected components (DCC) is marked in red.

$\mathbf{TS}_1$, $\mathbf{TS}_2$ and $\mathbf{TS}_4$, where the network of 100 nodes reached 35% of observability. In contrast, these rates remain slightly decreased for the large networks, with the exception of the threat scenarios based on $\mathbf{TS}_1$ and $\mathbf{TS}_2$, as shown in Table 3. This means that structural observability is influenced not only by the structure of driver nodes or their dependent nodes, but

```

profile on
clear all

%%% To generate a directed ER graph, two boundaries n and p must be
      defined, where n is the number of vertices and p is the edge
     probability %%%

n=input('Enter Num of Nodes (n)=  ')
pro=input('Enter Probability (p)= ')

num_e=pro*n*(n-1)/2;
num_e=round(num_e);
r=zeros(n);
IND=randperm(n*n,n*n);
y=0;
      for u=1:length(IND)
      [I,J] = ind2sub([n,n],IND(u));
      Li=sub2ind([n,n],J,I);
        y(u)=IND(u);
        y(y==Li)=[];
        if length(y(y~=0))==num_e
           break;
        end
      end

e=y(y~=0);

for  i=1:num_e
   [I,J] = ind2sub([n,n],e(i));
   r(I,J)=1;
end

%%% The Original Graph before Computation PDS %%%

g=digraph(r);
plot(g,'layout','force')

%%% Removing Isolated Nodes %%%

for i=1:n
    iso_n(i,1)=sum(r(i,:))+sum(r(:,i));
end
  iso_n_num=find(0==iso_n);
  new_g=rmnode(g,transpose(iso_n_num));
  new_r= full(adjacency(new_g));
  nn=size(new_r);n=nn(1,1);

if num_e>0
rr=new_r;
gg=new_g;
a=1

while sum(sum(rr))>0
for i=1:n
    emax(i,1)=sum(rr(i,:));
end

pp=find(emax==max(emax))
p=pp(1);
if max(emax)==1 && length(pp)>1
pg_ranks = centrality(gg,'outcloseness');
p=find(pg_ranks==max(pg_ranks))
p=p(1)
end

s=find(rr(p,:)~=0)
powe{a}=p;
se{a}=s;
rr(:,p)=0;
rr(p,s)=0;
rr(:,s)=0;
gg=digraph(rr)
for i=1:length(s)
 nn_in = nearest(gg,s(i),Inf)
 d=distances(gg,s(i),nn_in)
 dd=find(d==max(d))

```

Pseudocode 1. **The code illustrates the process of computing PDS for a given directed ER graph step by step, with the results of the computational simulation for a network of 500 nodes shown in Figure 2.**

```
 if isempty(dd)==0
    [mf,GF] = maxflow(gg,s(i),nn_in(dd(1)),'augmentpath')
    if mf==1

   m{a,i}= GF.Edges{:,1}
   u=unique(m{a,i})
   rr(u,:)=0;
   rr(:,u)=0;
   gg=digraph(rr);
    else  m{a,i}=[];
     rr(s(i),:)=0;
     gg=digraph(rr);
    end
 else
     m{a,i}=[];
     rr(:,s(i))=0;
     gg=digraph(rr);
 end
end
 a=a+1;
end

m=reshape(m,[],1);
m=cell2mat(m);
z=0;
for i=1:length(powe)
  for j=1:length(se{1,i})
      x(1,j+z)=powe{1,i}
      x(2,j+z)=se{1,i}(1,j)
  end
  z=length(x);
end

x1=nonzeros(x(1,:));x2=nonzeros(x(2,:));
x=[x1,x2];
output=cat(1,x,m)

for i=1:n
    if isempty(find(output==i))==1
         v(i)=i;
    else v(i)=0;
    end
end

v=transpose(nonzeros(v));
power_node=cat(2,cell2mat(powe),v);
s_n=cell2mat(se);
num_of_powernode=length(power_node);
cn=unique(output);
for i=1:num_of_powernode
    cn(find(cn==power_node(i)))=[]
end
controlled_node=cn;
num_of_cn=length(controlled_node);

%%% Representing the Whole Graph with Each PDS (i.e. Driver Nodes)
     and its Dependent Nodes %%%

g2=digraph();
g2=addnode(g2,n)
g2=addedge(g2,output(:,1),output(:,2))
num_of_edges=height(g2.Edges);
figure;
h2=plot(g2,'layout','force');
highlight(h2,power_node,'NodeColor',[0 1 0],'MarkerSize',4);

%%% The Final Graph after Computation PDS %%%

figure
h1=plot(new_g,'layout','force');
highlight(h1,power_node,'NodeColor',[0 1 0],'MarkerSize',4);
highlight(h1,output(:,1),output(:,2),'EdgeColor',[0 1 0],'LineWidth
     ',1.3)
else
figure;
h=plot(new_g,'layout','force');
highlight(h,[1:n],'NodeColor',[0 1 0]);
end
```

Pseudocode 1. ***(Continued.)*** **The code illustrates the process of computing PDS for a given directed ER graph step by step, with the results of the computational simulation for a network of 500 nodes shown in Figure 2.**

also by the behaviour of the attack scenario. As the networks are also attacked by $\mathbf{TS}_3$, their robustness is also evaluated. It is observed that this threat has no considerable effect on observability even for the network with a small number of nodes, where observability rates remain high (above 91%

```
%%% Applying Threat Scenarios, denoted by TS %%%
g3=g2
eff_n={};
selected_nodes=[];
num_uncnode=[];
num_redge=[];
num_rpnode=[];
eff_ns=[];

while(1)
    g3=g2
    eff_n={};
selected_nodes=[];
num_uncnode=[];
num_redge=[];
num_rpnode=[];
eff_ns=[];

sc=input('Enter Type of Scenario:[1,2,3,4] = Zero for Break')
if sc==0
    break;
end
num_exec=input('Enter Number of Executions = ');
exc_disp=input('Display Each Execution = 1_Yes 2_No ');
for i=1:num_exec
    switch sc
        case {1,2,3}
    pg_ranks = centrality(g3,'outcloseness');
    onlycn=zeros(1,length(pg_ranks));
    onlycn(power_node)= pg_ranks(power_node)
    if sum(onlycn)==0
        display('All Power Nodes are Disconnected')
         sel_n=0;
          conn_b= 0;dia_b=0;
     conn_af= 0;dia_af=0;
    else
        display('s1s2s3')
       maxpowern= find( onlycn==max(onlycn))
       maxpowern=maxpowern(1);
    [num_uncn,num_re,num_rpn,newgraph,sel_n,eff_node]=scenario(g3,
         maxpowern,power_node,sc)
     num_uncnode(length(num_uncnode)+1)=num_uncn;
     num_redge(length(num_redge)+1)=num_re;
     num_rpnode(length(num_rpnode)+1)=num_rpn;
     conn_b= outdegree(g3,sel_n);
     conn_af= outdegree(newgraph,sel_n);
     l=nearest(g3,sel_n,Inf);
   dia_b= max(distances(g3,sel_n,l));
   l2=nearest(newgraph,sel_n,Inf);
   dia_af= max(distances(newgraph,sel_n,l2));
   if isempty(dia_af)==1
       dia_af=0;
   end
   if sc==1 || dia_af==0
   dis_conncomp=num_redge+1;
   sum_disccomp=sum(num_redge)+1;
   else
       dis_conncomp=num_redge;
       sum_disccomp=sum(num_redge);
   end
   sum_disccomp=sum(dis_conncomp);
g3=newgraph
    end
    case 4
pg_ranks = centrality(g3,'outcloseness')
for n=1:length(power_node)
sucIDs{n} = successors(g3,power_node(n))
end
sucID=cell2mat(reshape(sucIDs,[],1))
nodes=find(pg_ranks~=0)
    onlycn=zeros(1,length(pg_ranks))
    rms5n=find(pg_ranks==min(pg_ranks(nodes)))
    onlycn(nodes)= pg_ranks(nodes)
    for k=1:length(rms5n)
        if ismember(rms5n(k),sucID)==0
    onlycn(rms5n(k))= 0
        end
    end
```

Pseudocode 2. **The main code for running the vulnerability scenarios (TS_1, TS_2, TS_3 and TS_4). Note that Pseudocodes 1 and 2 should be combined into a single Matlab file, with the latter placed after Pseudocodes 1.**

```
    onlycn(eff_ns)=0
    onlycn(power_node)= 0
    if sum(onlycn)==0
        display('All Target Nodes are Disconnected')
      sel_n=0;
       dia_b=[];dia_af=[];
    else
          display('s4')
        target_node=find(onlycn~=0)
    [num_uncn,num_re,num_rpn,newgraph,sel_n,eff_node]=scenario(g3,
         power_node,target_node,sc)
      num_uncnode(length(num_uncnode)+1)=num_uncn;
     num_redge(length(num_redge)+1)=num_re;
     sel_pns4=0;
     for j=1:length(power_node)
     controlled_nodes{j}=nearest(g3,power_node(j),Inf);
    if ismember(sel_n,controlled_nodes{j})==1 & sel_n~=0
        sel_pns4=power_node(j) ;
    end
     end
     if sel_pns4==0 || sel_n==0
         dia_b=0;dia_af=0;
     else
     l=nearest(g3,sel_pns4,Inf);
   dia_b= max(distances(g3,sel_pns4,l));
   l2=nearest(newgraph,sel_pns4,Inf);
   dia_af= max(distances(newgraph,sel_pns4,l2));
     end
      dis_conncomp=num_redge;
       sum_disccomp=sum(num_redge);
        sum_disccomp=sum(dis_conncomp);
g3=newgraph
    end
    end
eff_n{length(eff_n)+1}=eff_node;
eff_ns=cat(1,eff_n{1,:});
if sel_n ~=0
selected_nodes(length(selected_nodes)+1)=sel_n;
end
selected_nodes
num_uncnodes=length(unique(eff_ns));
num_redges=sum(num_redge);
 num_rpnodes=sum(num_rpnode);
 num_selected_n=length(unique(selected_nodes));
 sum_selected_nrep=length(selected_nodes);
 pg_ranks = centrality(g3,'outcloseness');
 u=ismember(power_node,selected_nodes);
 u1=power_node(find(u==1));
 u2=find(pg_ranks(u1)==0);
 u3=find(pg_ranks(u1)~=0);
 Orang_pn=u1(u3);
 red_pn=u1(u2);
 if exc_disp==1
figure;
 end
 if sc==4
tit= [num2str(i), ' dia [',num2str(dia_b),',', num2str(dia_af),' ]
      discc[',num2str(dis_conncomp),']','sum_dcc[ '...
   ,num2str(sum_disccomp),' ]'];
 else if sc==5
         tit=[num2str(i)];
     else
     tit= [num2str(i),' conn [',num2str(conn_b),',',num2str(conn_af
           ),']'...
    ' dia [',num2str(dia_b),',', num2str(dia_af),' ] discc[',
          num2str(dis_conncomp),']','sum_dcc[ '...
   ,num2str(sum_disccomp),' ]'];
     end
 end
    h2=plot(g3,'layout','force'); title( tit);

highlight(h2,power_node,'NodeColor',[0 1 0],'MarkerSize',4);
highlight(h2,selected_nodes,'NodeColor',[1 0.5 0.1 ])
highlight(h2,Orang_pn,'NodeColor',[1 0.5 0.1 ],'MarkerSize',5)
highlight(h2,eff_ns,'NodeColor','r')
highlight(h2,red_pn,'NodeColor','r','MarkerSize',5);
unobs_r=(num_uncnodes+num_rpnodes)/ height(g3.Nodes);
obs_r=((length(controlled_node)-num_uncnodes)+(num_of_powernode-
      num_rpnodes))/ height(g3.Nodes);
end
end
```

Pseudocode 2. ***(Continued.)*** **The main code for running the vulnerability scenarios (TS_1, TS_2, TS_3 and TS_4). Note that Pseudocodes 1 and 2 should be combined into a single Matlab file, with the latter placed after Pseudocodes 1.**

in the worst case) after an attack. Since the behaviour of $\mathbf{TS}_3$ targets only a few (but not all) edges of a vertex in N_D with the minimum out-degree. As shown in Table 3, which demonstrates observability rates, $\mathbf{TS}_4$ also causes only slight

```
function [num_uncn,num_re,num_rpn,newgraph,sel_n,eff_node]=scenario
      (g,power_node,all_p,sc)

%%% Applying Threat Scenarios TS1 %%%

if sc==1
num_uncn=length(nearest(g,power_node,Inf));
if num_uncn==0
    num_rpn=0;
else
    num_rpn=1;
end
 sucIDs = successors(g,power_node);
  num_re=length(sucIDs);
  rem_e(1:num_re,1)=power_node;
  newgraph=rmedge(g,rem_e,sucIDs);
  sel_n=power_node;
  eff_node=nearest(g,power_node,Inf);
end

%%% Applying Threat Scenarios TS2 %%%

if sc==2

    pg_ranks = centrality(g,'outcloseness');
    onlyp=zeros(1,length(pg_ranks));
    onlyp(all_p)= pg_ranks(all_p);
maxp=find(onlyp==max(onlyp))
maxpower=maxp(1)

    sucIDs = successors(g,maxpower)
    if length( sucIDs) > 1
    r=randi(length( sucIDs)-1,1,1)
    num_rpn=0;
    else r=length( sucIDs);  num_rpn=1;
    end
    r2=randperm(length( sucIDs),r)
    num_re=r
    selectededge=sucIDs (r2)
    p(1:r,1)=maxpower;
       newgraph=rmedge(g,p,selectededge)
       for i=1:r

           uncn{i}= nearest(g,selectededge(i,1),Inf)
       end
       num_uncn=length(cat(1,uncn{1,1:r}))+r
       sel_n=maxpower;
        eff_node=cat(1,selectededge,uncn{1,:})
end

%%% Applying Threat Scenarios TS3 %%%

if sc==3

 pg_ranks = centrality(g,'outcloseness');
    onlyp=zeros(1,length(pg_ranks));
    onlyp(all_p)= pg_ranks(all_p);
    if length(find(onlyp~=0))>1
        onlyp(power_node)=0;
    end
maxp=find(onlyp~=0)
maxpower=maxp(randi(length(maxp),1,1))

    sucIDs = successors(g,maxpower)
    if length( sucIDs) > 1
    r=randi(length( sucIDs)-1,1,1)
    num_rpn=0;
    else r=length( sucIDs);num_rpn=1;
    end
    r2=randperm(length( sucIDs),r)
    num_re=r
    selectededge=sucIDs (r2)
    p(1:r,1)=maxpower;
       newgraph=rmedge(g,p,selectededge)
       for i=1:r
           uncn{i}= nearest(g,selectededge(i,1),Inf)
       end
       num_uncn=length(cat(1,uncn{1,1:r}))+r
       sel_n=maxpower;
       eff_node=cat(1,selectededge,uncn{1,:})
end

%%% Applying Threat Scenarios TS4 %%%

if sc==4
maxp=all_p;
maxpower=maxp(randi(length(maxp),1,1))

    sucIDs = successors(g,maxpower)
    num_re=1
    selectededge=sucIDs ;
    newgraph=rmedge(g,  maxpower,selectededge)
  uncn= nearest(g,maxpower,Inf)
 num_uncn=length(uncn);
       num_rpn=0;
       sel_n=maxpower;
         eff_node=uncn;
end
end
```

Pseudocode 3. **To accomplish the threat scenarios, Pseudocode 2 calls the functions in this code. After saving the code to a separate file and renaming it "scenario" copy it in the same direction as the Matlab file.**

Pseudocode 3. ***(Continued.)*** **To accomplish the threat scenarios, Pseudocode 2 calls the functions in this code. After saving the code to a separate file and renaming it "scenario" copy it in the same direction as the Matlab file.**

damage to the structural control in most networks, with the exception of small networks. This occurs mainly because this attack removes at most one edge during each attack round from a vertex not belonging to the set of N_D. However, the observation degree of small networks under $\mathbf{TS}_4$ attacks decreases drastically due to their low connectivity probability, which produces a smaller number of edges joining each pair of vertices in the networks (i.e. any network has fewer connections between vertices).

The diameter of the network after removing edges is assessed to measure the robustness of network structural controllability against edge removals. This metric mainly relies on calculating the distance between two nodes in a network after an attack, particularly nodes belonging to driver nodes and their dependent nodes, and this is done by computing the number of edges in the shortest dependency path between such nodes. Table 3 shows that the networks begin to lose the control diameter values due to $\mathbf{TS}_1$ attacks, where the values reach null and become variable under threats of type $\mathbf{TS}_2$. In addition, the networks are only somewhat influenced by the attack scenarios $\mathbf{TS}_3$ and $\mathbf{TS}_4$; notably, in some cases the control network diameter has no change at all following the attacks.

For each attack scenario, the numbers of **AN** and **DCC** are computed when the networks are vulnerable to four attack models ($\mathbf{TS}_1$, $\mathbf{TS}_2$, $\mathbf{TS}_3$ and $\mathbf{TS}_4$). This computation allows to determine the minimum-size set of PDS necessary to control the compromised nodes in event of recovery of structural controllability, where the number of driver nodes needed to control the **AN** is equal to the size of **DCC**, as shown in Table 4. With $\mathbf{TS}_1$ and $\mathbf{TS}_2$, the fraction of **AN** in most networks continuously increases along with the number of **DCC**. As Table 4 and Figure 1 illustrate, the number of compromised nodes dramatically increases when more edges are removed during each attack round, leading to an increase in the **DCC** size required to achieve full control and, significantly, for networks under attack from $\mathbf{TS}_1$ and $\mathbf{TS}_2$. However, there is an insignificant variation in the number of **DCC** when the networks are subjected to $\mathbf{TS}_3$ and $\mathbf{TS}_4$; this variation depends on the nature of the attack $\mathbf{TS}_3$, which targets at least one edge or several (but not all) edges of a vertex within N_D of the minimum out-degree. In the case of $\mathbf{TS}_4$, the number of **DCC** required to monitor **AN** in each attack round is equal to one driver node at most, although the number of **AN** increases, as shown in Table 4. This is due to the fact that the behaviour of $\mathbf{TS}_4$ exploits one edge at most in each attack round, resulting in a minor impact on the size of **DCC**. Additionally, the position of the node to which the attacked edge belongs may be in the middle of a dependency path controlled by PDS (i.e. the compromised nodes are a descendant of the node to which its edge is removed), resulting in the complete isolation of a sequence of dependent nodes that are controlled by the attacked node.

V. CONCLUSION

Structural controllability provides an efficient graph-theoretical understanding of network structural properties and their critical elements in large cyber-physical control networks. This paper, therefore, focused on an alternative method based on the power dominating set problem to identify the minimum number of driver nodes which must be considered crucial for attackers attempting to compromise the network control.

The paper has discussed a simulation experiment analysing the robustness of structural controllability for directed ER networks and their power domination in terms of structural connectivity and observability when the networks were exposed to vulnerability attacks, particularly multi-round edge removals in various scenarios. The robustness of networks showed a unique behaviour when subject to threats of type $\mathbf{TS}_1$, $\mathbf{TS}_2$, $\mathbf{TS}_3$ and $\mathbf{TS}_4$. The simulation results demonstrated that $\mathbf{TS}_1$ has a significantly harmful effect on structural controllability as it enables adversaries to attack a considerable fraction of edges in the whole original network, leading to disrupting legitimate control. $\mathbf{TS}_2$ also poses a threat to the connectivity of the networks but is not dangerous. The results also highlighted that $\mathbf{TS}_1$ had a clear influence on the networks' control diameter values, while the networks became less affected by the attack scenarios $\mathbf{TS}_3$ and $\mathbf{TS}_4$.

The paper has also presented the disconnected components (**DCC**) to calculate a minimum set of N_D required to control the compromised nodes following an attack. The number of **DCC** dramatically increased when additional edges were removed in each attack round, leading to an increase in the **DCC** size required to gain full control.

Ongoing and prospective research focuses on the impact of such attacks on various networks and similar control topologies, mainly small-world (Watts-Strogatz) and scale-free (Barabási-Albert) graphs. In addition, a recovery algorithm will be developed to preserve network structural controllability in the presence of adversaries capable of removing power links partially.

APPENDIX

See Figure 2–6 and Pseudocode 1–3.

ACKNOWLEDGMENT

The research would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

REFERENCES

[1] C.-L. Pu, W.-J. Pei, and A. Michaelson, "Robustness analysis of network controllability," *Phys. A, Stat. Mech. Appl.*, vol. 391, no. 18, pp. 4420–4425, Sep. 2012.

[2] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 5, May 2002, Art. no. 056109.

[3] L. K. Gallos, R. Cohen, F. Liljeros, P. Argyrakis, A. Bunde, and S. Havlin, "Attack strategies on complex networks," in *Proc. 6th Int. Conf. Comput. Sci.* Berlin, Germany: Springer, 2006, pp. 1048–1055.

[4] A. Melchionna, J. Caloca, S. Squires, T. M. Antonsen, E. Ott, and M. Girvan, "Impact of imperfect information on network attack," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 91, no. 3, Mar. 2015, Art. no. 032807.

[5] C.-T. Lin, "Structural controllability," *IEEE Trans. Autom. Control*, vol. AC-19, no. 3, pp. 201–208, Jun. 1974.

[6] A. Olshevsky, "Minimal controllability problems," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 3, pp. 249–258, Sep. 2014.

[7] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 91–101, Mar. 2016.

[8] L. Deng, S. Fu, Y. Li, P. Zhu, and H. Liu, "Controllability and optimal control of higher-order incomplete Boolean control networks with impulsive effects," *IEEE Access*, vol. 6, pp. 71003–71011, 2018.

[9] R. E. Kalman, "Mathematical description of linear dynamical systems," *J. Soc. Ind. Appl. Math. A, Control*, vol. 1, no. 2, pp. 152–192, 1962.

[10] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, pp. 167–173, May 2011.

[11] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, "Domination in graphs applied to electric power networks," *SIAM J. Discrete Math.*, vol. 15, no. 4, pp. 519–529, Jan. 2002.

[12] W.-X. Wang, X. Ni, Y.-C. Lai, and C. Grebogi, "Optimizing controllability of complex networks by minimum structural perturbations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 85, no. 2, Feb. 2012, Art. no. 026115.

[13] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex networks," *Rev. Modern Phys.*, vol. 88, no. 3, 2016, Art. no. 035006.

[14] B. Y. Chan and R. D. Shachter, "Structural controllability and observability in influence diagrams," in *Proc. 8th Int. Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 1992, pp. 25–32.

[15] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith, "Parameterized power domination complexity," *Inf. Process. Lett.*, vol. 98, no. 4, pp. 145–149, May 2006.

[16] U. Feige, "A threshold of ln *n* for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998.

[17] J. Guo, R. Niedermeier, and D. Raible, "Improved algorithms and complexity results for power domination in graphs," *Algorithmica*, vol. 52, no. 2, pp. 177–202, Oct. 2008.

[18] A. Aazami and K. Stilp, "Approximation algorithms and hardness for domination with propagation," *SIAM J. Discrete Math.*, vol. 23, no. 3, pp. 1382–1399, Jan. 2009.

[19] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.

[20] M. Barthélemy, "Betweenness centrality in large complex networks," *Eur. Phys. J. B, Condens. Matter*, vol. 38, no. 2, pp. 163–168, Mar. 2004.

[21] B. Wang, L. Gao, Y. Gao, and Y. Deng, "Maintain the structural controllability under malicious attacks on directed networks," *Europhys. Lett.*, vol. 101, no. 5, pp. 1–6, 2013.

[22] A. E. Motter and Y.-C. Lai, "Cascade-based attacks on complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 66, no. 6, pp. 378–382, Dec. 2002.

[23] C. Alcaraz, E. E. Miciolino, and S. D. Wolthusen, "Structural controllability of networks for non-interactive adversarial vertex removal," in *Proc. 8th Int. Workshop Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2013, pp. 120–132.

[24] B. Alwasel and S. D. Wolthusen, "Reconstruction of structural controllability over Erdős-Rényi graphs via power dominating sets," in *Proc. 9th Annu. Cyber Inf. Secur. Res. Conf. (CISR)*. New York, NY, USA: ACM, 2014, pp. 57–60.
[25] B. Alwasel and S. D. Wolthusen, "Recovering structural controllability on Erdős-Rényi graphs via partial control structure re-use," in *Proc. 9th Int. Conf. Crit. Inf. Infrastruct. Secur.* Cham, Switzerland: Springer, 2014, pp. 293–307.
[26] B. Alwasel and S. D. Wolthusen, "Recovering structural controllability on Erdős-Rényi graphs in the presence of compromised nodes," in *Proc. 10th Int. Conf. Crit. Inf. Infrastruct. Secur.* Springer, 2015, pp. 105–119.
[27] C. Alcaraz and S. D. Wolthusen, "Recovery of structural controllability for control systems," in *Proc. 8th Int. Conf. Crit. Infrastruct. Protection*. Berlin, Germany: Springer, 2014, pp. 47–63.
[28] S. Zhang and S. D. Wolthusen, "Iterative recovery of controllability via maximum matching," in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Aug. 2017, pp. 328–333.
[29] S. Zhang and S. D. Wolthusen, "Structural controllability recovery via the minimum-edge addition," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 5822–5827.
[30] S. Zhang and S. D. Wolthusen, "Driver-node based security analysis for network controllability," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 2246–2251.
[31] B. Bollobás, *Random Graphs*. Cambridge, U.K.: Cambridge Univ. Press, 2001.
[32] B. Alwasel, "Recovery of structural controllability into critical infrastructures under malicious attacks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 723–728, 2020.

B. ALWASEL received the B.S. degree in computer science from King Saud University, Qassim Branch, Saudi Arabia, in 2004, the M.S. degree in computer security from the University of Birmingham, Birmingham, U.K., in 2011, and the Ph.D. degree in information security from Royal Holloway, University of London, London, U.K., in 2016.

Since 2016, he has been an Assistant Professor with the Department of Applied Natural Science, Qassim University. Since 2018, he has been the head of the department. His main research interests include cyber-physical systems security, network and distributed systems security, control systems, and graph theory and models for critical infrastructure protection.