



## OPEN ACCESS

## EDITED BY

Christian Beck,  
Queen Mary University of London,  
United Kingdom

## REVIEWED BY

Maxi San Miguel,  
IFISC (CSIC-UIB), Spain  
Claudio Castellano,  
Istituto dei Sistemi Complessi (ISC-CNR), Italy

## \*CORRESPONDENCE

Vesa Kuikka,  
✉ vesa.kuikka@aalto.fi

RECEIVED 29 April 2025

ACCEPTED 26 August 2025

PUBLISHED 16 September 2025

## CITATION

Kuikka V, Pykälä L, Takko T and Kaski KK (2025)  
Network modelling in analysing cyber-  
related graphs.  
*Front. Complex Syst.* 3:1620260.  
doi: 10.3389/fcpxs.2025.1620260

## COPYRIGHT

© 2025 Kuikka, Pykälä, Takko and Kaski. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](#). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Network modelling in analysing cyber-related graphs

Vesa Kuikka \*, Lauri Pykälä, Tuomas Takko and Kimmo K. Kaski

Department of Computer Science, Aalto University School of Science, Otakaari, Finland

To improve the resilience of the computer network infrastructure against cyber attacks or causal influences and find ways to mitigate their impact, we need to understand their structure and dynamics. Here, we propose a novel network-based influence-spreading modelling approach to investigate event trajectories or paths in attack and causal graphs with directed, weighted, cyclic and/or acyclic paths. In our model, we can perform probabilistic analyses that extend beyond traditional methods to visualise cyber-related graphs. The model uses a probabilistic method to combine paths that join within the graph. This analysis includes vulnerabilities, services, and exploitabilities. To demonstrate the applicability of our model, we present three cyber-related use cases: two attack graphs and one causal graph. This model can serve cyber analysts as a tool to produce quantitative metrics for prioritising tasks, summarising statistics, or analysing large-scale graphs.

## KEYWORDS

attack modelling technique, network modelling, attack graph, causal graph, directed acyclic graph

## 1 Introduction

Cyber attacks are unauthorised actions against computer network infrastructure to exploit vulnerabilities in target systems, while cyber threats refer to potential adversarial events in which an attacker could exploit vulnerabilities (Li and Liu, 2021). Typically, a cyber attack can consist of a number of actions, events, or steps required for the attacker to gain access or sufficient privileges to achieve the objective of an attack, e.g., denial of service (DOS) or the delivery and use of malware. A common approach to classifying these steps has been through various frameworks such as the cyber kill chain (Hutchins et al., 2011). At a more technical level, potential or real cyber attacks have often been modelled as directed graph structures. Among these graph-based methods, attack graphs and their variants are often used to represent cyber attacks (Lallie et al., 2020; Liu et al., 2012; Wachter, 2023; Zeng et al., 2019; Zenitani, 2023). There are two commonly used forms of attack graphs: the first is a directed graph with nodes representing network states and edges (links) representing exploits, while the second uses nodes to represent pre- or post-conditions of an exploit and edges to show the consequences. In addition, knowledge-based methods are used to model attack concepts and vulnerability details (Alserhani, 2015; Qi et al., 2023; Sikos, 2023).

In order to understand, analyse, and visualise the sequence of events of a successful cyber attack, we will here use attack modelling techniques (AMTs). There are three main categories of AMT: (i) methods based on use cases like misuse or security use cases, (ii) temporal methods based on, e.g., cyber kill chains, and (iii) methods based on graphs (Lallie et al., 2020). Here, we will focus on graph-based models as they enable cyber analysts and researchers to formulate and use different metrics, both quantitative and qualitative, to optimise and prioritise defensive efforts (Lallie et al., 2020). Graph-based cyber attack modelling can be conducted at various

technological levels, including applications, services, or entire computer networks. To achieve this, mappings from technical micro-service levels to higher clustering levels can be accomplished using technical configuration repositories.

In the literature, the term *attack graph* has been used to refer to a variety of modelling approaches (Wachter, 2023), which present a taxonomy of these models based on analysing 70 attack graph formalisms. This taxonomy uses a hierarchical categorisation with two top-level categories: sequential models and causal dependency models. Sequential models represent attack steps and paths formed by them, while causal dependency models focus on cause-effect relationships. Based on node semantics, sequential models are further categorised into asset-oriented, vulnerability-/exploit-oriented, and condition-/attribute-oriented types.

Our research approach will focus on network modelling, as it provides a range of methods and tools for analysing cyber-related graphs from different perspectives and aggregation levels. One recent approach involves modelling network structure and influence flow at a detailed level of nodes and links (Almiola et al., 2023; Kuikka and Kaski, 2024; Kuikka et al., 2022). This network model uses Markov processes and applies probability theory to assess the combined effects from multiple sources. Our method for evaluating the effects of various joining paths in both acyclic and cyclic (Levner and Tsadikovich, 2024) graphs extends the probabilistic techniques discussed in the literature (Carter et al., 2014; Homer et al., 2013; Stergiopoulos et al., 2022; Wang et al., 2008). In this study, we calculate the combined effect of multiple alternative paths in a graph by using the theorem of non-mutually exclusive events from probability theory. Details of the model and its corresponding algorithm can be found in (Kuikka et al., 2022). In general, our probabilistic modelling approach serves as an integrated, graph-based methodology or framework for analysing various types of influence spreading processes, including cyber attacks. The integration of modelling and analysis metrics within the same probabilistic framework makes our approach self-consistent, distinguishing it from other graph-based methods.

In this study, we present the modelling of cyber-related graphs through three use cases and demonstrate how the model can be applied to examine the exploitability, causality, and incomplete graph information in the computer network and services system. The first use case (Stergiopoulos et al., 2022) involves a directed acyclic graph in which the states of the system are represented as nodes and the vulnerabilities as links between the nodes, forming an attack graph. The exploitability of a vulnerability is interpreted as the probability of a successful exploit (i.e., traversability of a link). In the network model, these probabilities are assigned to link weights, which are derived using general CVSS scoring data (Common Vulnerability Scoring System) (NIST, 2012) and adjusted with a scaling factor. The second use case (Alsaheel et al., 2021) demonstrates a larger network structure of services and applications, illustrating how exploitability metrics can be utilised to produce aggregate security metrics concerning vulnerability and service levels. The third use case (Alsaheel et al., 2021; Jiang et al., 2017) involves two types of nodes, namely, attack entities and non-attack entities identified by an attack investigation tool using natural language processing and deep learning. This use case illustrates how our model can address inaccuracies in results from threat analysis tools.

These use cases demonstrate how different categorisations can be applied within the model. Because the model presents metrics as probabilities, it aids in forming a consistent understanding of the situation from various perspectives. Additionally, our model introduces novel methods for analysing cyber-related graphs. It employs a probabilistic technique to merge multiple alternative paths—both cyclic and acyclic—within a graph. The model uses a detailed model to consider all potential paths through the nodes of the network and defines new metrics to estimate the exploitability and impacts. For instance, it can be used to evaluate the impact of eliminating certain vulnerabilities across services or protecting targeted services within the system. Therefore, our model could serve as a valuable aid for cyber analysts and those designing cyber-secure systems. Based on our model, it is possible to develop a visualisation and analysis tool that allows cyber analysts to leverage various model outputs and metrics for real-world mitigation planning. This tool could include enterprise-specific assessments related to exploitability and impact values. As a result, analysts could perform what-if analyses and inspect specific parts of networks and systems.

In Section 3, we will present our network modelling method and analysis metrics. The pseudo-algorithms of our influence spreading model are reproduced in the supplementary material from our earlier study (Kuikka et al., 2022). In Section 2, we review related research on attack graphs and metrics used to assess the impacts of cyber attacks. In Section 4, we describe the data and networks of the use cases on which the model is demonstrated. In Section 5, we will explain how the model can be used to obtain new results in analysing cyber-related graphs through the three use cases. In Section 6, we will discuss the applications of our graph-based methods in analysing cyber vulnerabilities and attacks, and present concluding remarks.

## 2 Related work

Several reviews have been published about attack graphs and related cybersecurity methods and models to analyse them (Lallie et al., 2020; Wachter, 2023; Zeng et al., 2019; Zenitani, 2023). The review in (Lallie et al., 2020) provides a description of a theory of cyber attacks and how elements of attack graphs and attack trees are represented and visualised. The study in (Wachter, 2023) presents the state of research on the representation and analysis of cyber attacks using attack graph formalisms and proposing a classification based on an analysis of models regarding graph semantics, agents involved, and analytical features. This study also investigates which formalisms enable automatic attack graph generation from raw or processed data input.

The review in (Zenitani, 2023) presents a recent summary of studies on attack graphs, by focusing on key concepts such as exploit dependency and/or rules (Levner and Tsadikovich, 2024), monotonicity, and handling cycles. The study in (Zeng et al., 2019) describes the key concepts, generation methods, and computation tasks of attack graphs. Also, it summarises various analysis methods, including graph-based, Bayesian network-based, Markov model-based, cost-optimisation, and uncertainty analysis methods. In the attack graph analysis, there is a need to quantify the levels of threats, vulnerabilities,

exploits, and impact, each with appropriate metrics. For example, in enterprise environments, system security metrics (Pendleton et al., 2016) and network attack graph metrics (Noel and Jajodia, 2017) have been proposed to evaluate and compare the threats and impacts of cyber attacks.

In the literature (Shakarian et al., 2015), an Independent Cascade (IC) model has been proposed as a way to describe influence maximisation. This model assumes that nodes activate their neighbours independently on the basis of a fixed probability rather than multi-step logical relationships, as in the case of our influence spreading model. A key difference is that the IC model does not consider joining paths, like our present model does by following the probabilistic rule of mutually non-exclusive events. This can result in a more realistic evaluation of the overall risk for the enterprise network and services. Our probabilistic method also allows us to calculate expected cumulative and non-cumulative impact values, as well as the circular effects of attack propagation.

The present study is related to two earlier works: one by Stergiopoulos et al. (2022) and another by Alsaheel et al. (2021). The data for the first two use cases are adapted from (Stergiopoulos et al., 2022), while the data for the third use case comes from (Alsaheel et al., 2021). The study in (Alsaheel et al., 2021) presents a sequence-based learning approach to investigate attacks, whereas (Stergiopoulos et al., 2022) proposes an automatic analysis of attack graphs to facilitate risk mitigation and prioritisation using probabilistic methods. Our method uses probability-based path combination, which complements the approach outlined in (Stergiopoulos et al., 2022). In that approach, single paths, as well as those with minimum and maximum exploitability and impact values, are assessed. However, our method is well suited for the comprehensive evaluation of enterprise networks. In contrast, the single path method allows for a detailed analysis of individual attack paths. Both techniques can be used together to examine different aspects of cyber attacks. Moreover, in enterprise-scale networks, analysing single paths can be computationally resource-intensive due to the vast number of potential combinations. Relying solely on subjective judgment to limit the set of possible paths may not produce accurate results.

Although there are a number of studies using graph-based cyber attacks modelling, a comprehensive approach applicable to various levels of applications and services in enterprise networks is still lacking. In the present study, we introduce a probabilistic method to quantitatively analyse cyber-related graphs and identify strategies to improve the resilience of network infrastructure.

### 3 Attack graph model

Graphs related to cyber security, such as attack graphs and causal graphs, are mainly created to help visualise complexities of attack sequences and plan ways to prevent them (Lallie et al., 2020). These graphs are usually generated from large volumes of data for this purpose (Sheyner and Wing, 2003; Stojanović et al., 2020). Consequently, the resulting graphs typically contain a small number of nodes, ranging from a few to hundreds.

In our network-based modelling approach (Kuikka et al., 2022) we consider and analyse cyber attack processes from the point of view of influence spreading for finding ways to prevent or mitigate potential or ongoing attacks (Haque et al., 2021; Segovia-Ferreira et al., 2024). The model describes the network structure, including individual nodes and links. Influence spreads through a non-conserved process, moving from a node to all neighbouring nodes via directed links. In our model, the influence spreading can be assumed to take place via acyclic or cyclic Markov processes on a complex network structure. In the present study, the model describes attack propagation along self-avoiding paths, representing acyclic paths, and uses unrestricted paths to represent general propagation through either acyclic or cyclic paths.

The weights assigned to directed links represent the probabilities of successful attack propagation steps within a graph structure. Since the actual parameter values, including link weights and CVSS scores, can vary greatly in real environments, in this study, we employ a scaling factor to cover the full range of possible parameter values. This approach also enables what-if and sensitivity analyses to explore various attack scenarios and environments.

Our model is quite versatile as it offers different approaches to dealing with full breakthrough of nodes or self-avoiding paths. The first approach allows loops in the spreading process, while the second approach is better for examining connectivity between nodes. Loops consisting of a minimum of three nodes connected by links enable circular propagation of influence within the network structure. Both approaches yield similar results for directed acyclic graphs because the network lacks cyclic processes. In case of full breakthrough effects, a bidirectional link creates a cycle between the two nodes at each end of the link, resulting in recurring events in the network. Note that due to spreading, the effects are not limited to those nodes but spread throughout the network structure. In the case of self-avoiding paths, no recurring effects are permitted, and the process can propagate only in one direction along a path.

The method and corresponding algorithms proposed in our earlier article (Kuikka et al., 2022) are based on modelling network structure and spreading processes. This results in an  $N \times N$  probability matrix  $C$ , where the number of nodes in the network is denoted by  $N$ . The matrix elements represent directed spreading or connectivity probabilities from one node to another in the network with a connecting path between them. When spreading from branching paths, it occurs in all possible directions, and the probabilities concern the first arrival at the end nodes. Subsequently, this matrix is used to define various metrics for analysing the network structure and network flow process. Note that models other than our influence spreading model can generate the probability matrix. As the analysis is based on the probability matrix, it is independent of the method used to produce the probability matrix. These models could even be non-Markovian or include other event types besides mutually non-exclusive events (OR rule) used to combine multiple joining alternative paths (Kuikka et al., 2022).

In network analysis, we use two metrics to evaluate the importance of a node, namely, out-centrality and in-centrality, and they are defined differently from the standard closeness centrality measures in the literature. One reason for using different metrics is the limitations of standard metrics in describing detailed network structures, because they are defined

based on the shortest paths between nodes, thus ignoring other alternative paths an attack could propagate (Landherr et al., 2010). We use the probability matrix as the basis of analysis, as it enables the separation of the network modelling from the analysis (Kuikka et al., 2022).

Next, we define the centrality measures for the subset of nodes  $V$  in the network  $G$  of  $N$  nodes. (Note that in the case of the entire network  $V = G$ ). For the influence spreading through the network, we define the out-centrality of the source node  $s$  by

$$C^{(out)}(s) = \frac{1}{N-1} \sum_{\substack{t \in V \\ s \neq t}} C(s, t) \tag{1}$$

and the in-centrality of a target node  $t$  by

$$C^{(in)}(t) = \frac{1}{N-1} \sum_{\substack{s \in V \\ s \neq t}} C(s, t). \tag{2}$$

Here, the out-centrality of a node is the average value of probabilities of spreading from the source node to all other nodes in  $V$ , while the in-centrality of a node is the average value of probabilities of spreading from all other nodes in  $V$  to the target node. These centrality measures describe the physical properties of the network structure and the flow process. As  $C(s, t)$  is the probability of influence spreading from node  $s$  to node  $t$ , the sum in Equation 1 is interpreted as the expected value of the influenced nodes and the sum in Equation 2 as the expected number of nodes that spread influence to the specified node. In the context of cyber-related graphs, influence spreading refers to the propagation of cyber attacks within the graph. In this study, we adopt the convention of setting the state value of the source node as zero and using the corresponding normalisation factor  $1/(N - 1)$  and express the results as percentage values.

When dealing with cyber-related graphs, the out-centrality is a measure for assessing the potential effect of a cyber attack on other nodes within the network. On the other hand, in-centrality is useful for understanding how various cyber attacks can affect nodes in the network. Detailed information can be obtained by focusing on a subset of start and end nodes using Equations 1, 2. Typically, the main focus is to explore vulnerabilities and the effect of a start node on an end node. However, multiple start nodes and end nodes can exist in one cyber-related graph.

Let us move on to discuss the metrics of exploitability and impact. The exploitability represents the probability of a successful cyber attack through a link, while impact describes its effect on the system, functionality, or operation. To assess the impact of different needs both at the technical and operational levels, the first step is to define the concept in the specific situation. The impact is calculated for the nodes and the exploitability is calculated for the links. We assume that the impact value does not influence the probabilities of propagation of the cyber attack. Therefore, impact values may not be expressed as probabilities and can have numerical values beyond the range of  $[0, 1]$ . However, the impact values for different nodes are comparable, enabling us to calculate the desired sums for paths or network structures.

When assessing vulnerabilities, there are two main approaches: the General CVSS scoring (Common Vulnerability

Scoring System) (NIST, 2012) or a custom evaluation method in which organisations supplement or override CVSS with their own risk assessments. The empirical CVSS scoring system provides a quantitative way to capture key characteristics of a vulnerability, resulting in scores that indicate its severity (the impact on a system) and exploitability (the ease of exploitation) (Mell et al., 2007). These metrics adhere to international standards for measuring cybersecurity risks. By using either CVSS or a custom evaluation, organisations can prioritise patches and mitigation strategies based on the potential impact of each vulnerability. We demonstrate the use of the CVSS scoring system in our first use case. In our second use case, since the general CVSS scores are uniformly high across all vulnerabilities, we focus on illustrating the network structure of services and applications and, therefore, use equal values for the vulnerability characteristics of all vulnerabilities. Furthermore, it is important to note that the general CVSS method may not accurately reflect the actual risk in a specific organisation or environment. The equations and algorithms used to score the base, temporal, and environmental metric groups are detailed in (Mell et al., 2007). Additional information on the origin and testing of these equations can be found at (FIRST, 2025).

Our approach is based on the use of link weights as probabilities to represent successful exploits. The exploitability metric measures the current state of exploit techniques or code availability. When easily accessible exploit code is publicly available, it increases the number of potential attackers, including unskilled ones, increasing the severity of vulnerabilities. The following equation illustrates how the factors *AccessVector*, *AccessComplexity*, and *Authentication* describe the accessibility and complexity of the vulnerability, and whether additional conditions are needed to exploit it. The exploitability metric is defined as follows (Mell et al., 2007):

$$Exploitability = 20 * AccessVector * AccessComplexity * Authentication. \tag{3}$$

Impact metric measures the potential consequences of exploiting a vulnerability of an IT asset. These impacts are independently defined based on the degree of loss in three key areas, namely, confidentiality, integrity, and availability. The present analysis can be further extended to consider specific business impacts by incorporating relevant effects in Equation 5. The total impact metric is defined as follows (Mell et al., 2007):

$$Impact = 10.41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact)). \tag{4}$$

In a network structure, a node can be targeted by different cyber attack paths from the source node through various links directed to the target node. In an attack graph, these may involve the same vulnerability. However, if different vulnerabilities can be exploited on one node, we calculate the weighted average value for the node or use the maximum impact of alternative exploits. We define a vector  $I$  with elements representing the impact values for the nodes in the network. The impact of a cyber attack from the start nodes  $S$  to the end nodes  $T$  can be defined as:



TABLE 1 Summary of the use case graphs.

Graph	Description	Nodes	Links	References
Multi-cloud Enterprise Network	Netflix OSS microservice system	18	25	<a href="#">Stergiopoulos et al. (2022)</a>
Netflix OSS	Netflix OSS microservice system	21	94	<a href="#">Stergiopoulos et al. (2022)</a>
ATLAS, Pony campaign attack	Recovered sequences and a causal graph	39	65	<a href="#">Alsaheel et al., 2021</a> , <a href="#">Jiang et al., 2017</a>

$$\mathcal{I}(S, T; I) = \sum_{\substack{i \in S \\ j \in T}} C(i, j) I_j, \tag{5}$$

where the matrix  $C$  consists of elements  $C(i, j)$  that represent the probabilities of successful attacks from node  $i$  to node  $j$  via alternative paths. The impact value on the end node alone includes the last exploit on the path. The impact value of Equation 5 can be calculated for specific events, including effects on specific services or aspects of security. Additionally, Equation 5 can be utilised to analyse the impact of exploits by determining the decrease in the impact value after mitigating vulnerabilities.

When all elements of vector  $I$  are set to one, we obtain a quantity that describes the attack propagation from the source node set  $S$  to the target node set  $T$  in the network. In this way, the impact value in Equation 5 is linked to the out- and in-centrality metrics in Equations 1, 2. Therefore, the centrality measures and the impact measure in Equation 5 are defined consistently with each other. Combining the alternative paths using probabilistic methods allows us to calculate accurate metrics based on Equations 3 and 4 for real-world cyber attacks and different classifications.

As a further characteristic measure, one can calculate a cumulative impact for an individual attack path, as done in (Stergiopoulos et al., 2022). In our model approach, the cumulative impact along a path is calculated by summing up the products of propagation probabilities and node impacts according to Equation 5. Once again, the impact on the end node only includes the impact on that node.

In summary, four kinds of impact metrics can be calculated: cumulative impacts along individual paths or all alternative paths, and non-cumulative impacts on the end nodes through individual or all alternative paths. In this study, we prefer metrics that combine alternative paths using a probability theory approach and methods. The factors  $C_{i,j}$  in Equation 5 account for the effects of all potential alternative paths. This approach allows for the definition of new metrics in analysing cyber events from different perspectives and categorisations. In the next section, we provide examples of this through use cases.

## 4 Use case data

For this study, we chose three different graph structures, described in Table 1 to serve as our use cases.

### 4.1 Use case 1: Multi-cloud enterprise network

The graph for the Multi-cloud Enterprise Network from (Stergiopoulos et al., 2022) represents the network topology of

two cloud infrastructures connected to the Internet behind an external firewall. The first cloud server hosts three virtual machines, all connected to a virtual switch. The second cloud server hosts a public network and a private network. External users can access a web server, and internal users can access the SQL server from inside the local network. Each server in this topology represents a realistic vendor-specific system with a set of real-world CVE vulnerabilities, with the impacts of exploitation extracted from the CVE database.

### 4.2 Use case 2: netflix OSS architecture

Ibrahim et al. (2019) used a combination of tools to construct attack graphs for microservice architectures, consisting of a Docker host running various interconnected containers. The network topology was extracted from the docker-compose configuration files that define the orchestration of the services and details about the connections between containers, published ports, and any privileged access granted to containers. Vulnerabilities were scanned from the Docker images using the vulnerability scanning software Clair, producing a list of CVEs with textual descriptions and attack vectors for each image. The topology and vulnerability data were used as input for the final attack graph generation process. Attack pre- and postcondition parsing was performed by matching the attack vectors of the vulnerabilities by keyword matching (Aksu et al., 2018).

### 4.3 Use case 3: pony APT (advanced persistent threat) campaign

In the research conducted on the Pony APT campaign (CVE-2017-0199) (Jiang et al., 2017), the methodology was based on the work by Alsaheel et al. (2021), describing a framework for constructing "attack stories" from audit logs. Their approach involved several steps. First, a platform independent causal graph representation was constructed using various system audit logs, including DNS records, web objects, processes, file accesses, and network connections. This graph was then abstracted using optimisation techniques to reduce its complexity. Specifically, nodes and edges that were not reachable from the attack nodes or attack symptom nodes were removed, repeated edges between entities were reduced to only the first occurrence, and nodes and edges referring to the same type of event were combined.

Next, the process of attack sequence construction took place. All "attack entities" from the causal graph were obtained, and subsets consisting of two or more attack entities were formed.

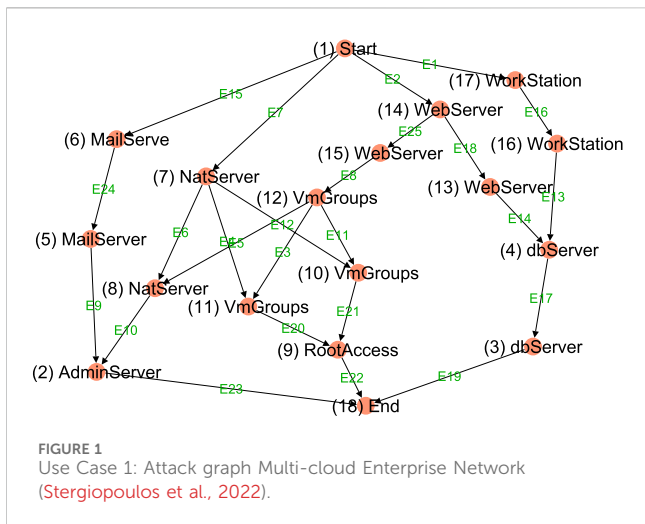


TABLE 2 Exploitability and impact values calculated from Equations 3, 4 for vulnerabilities in the attack graph of Figure 1 (Stergiopoulos et al., 2022).

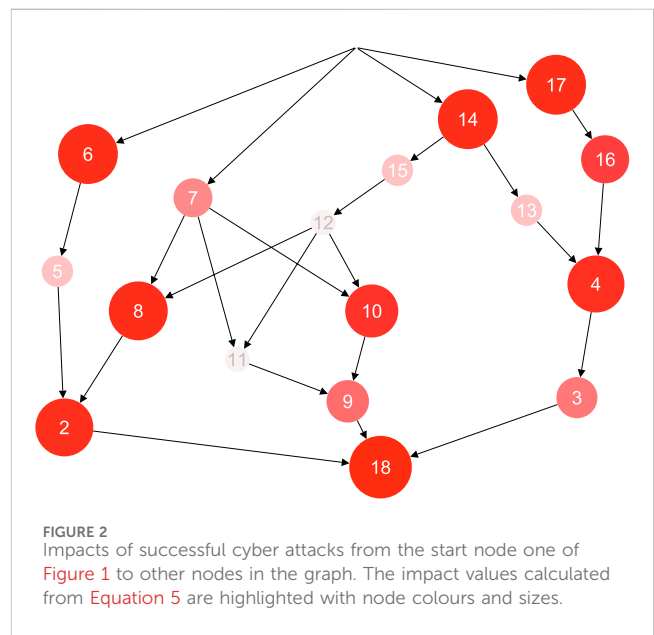
Vulnerability	Expl./10	Impact	Links
-	1	-	E1,E2
CVE-2010-3847	0.339258	10.00085	E3,E4,E20
CVE-2003-0693	0.99968	10.00085	E5,E6
CVE-2007-4752	0.99968	6.442977	E7
CVE-2001-0439	0.99968	6.442977	E8
CVE-2008-4050	0.85888	10.00085	E9,E10
CVE-2008-0015	0.85888	10.00085	E11,E12,E21, E22,E23
CVE-2009-1918	0.99968	10.00085	E13,E16
CVE-2018-7841	0.99968	6.442977	E14,E18
CVE-2004-0840	0.99968	10.00085	E15
CVE-2008-5416	0.7952	10.00085	E17,E19
CVE-2001-1030	0.99968	6.442977	E24
CVE-2009-1535	0.99968	6.442977	E25

The neighbourhood graph for each entity within these subsets was extracted, followed by retrieving time-stamped events for each neighbourhood graph. Sequences were labelled attack sequences if they consisted exclusively of attack events, which means that both the source and destination nodes were associated with attacks. Following this, sequence lemmatisation was performed. This involved transforming sequences into a textual representation using a general vocabulary of 30 words, divided into four types: process, file, network, and actions. To balance the dataset, non-attack sequences were undersampled, and attack sequences were oversampled by mutating existing attack sequences. Finally, the sequences were embedded and modelled, followed by an investigation of attack and recovery of “attack stories”. In the resulting processed graph, the semantic interpretations of nodes and edges are connected to system states inferred from the audit logs. The nodes represent processes (i.e., executables), files or IP addresses. The edges between them in turn represent actions or events between the entities, like read, write, execute, or connect.

The ATLAS approach thus focuses on reconstructing the attack story from observed system behaviours, with nodes and edges closely tied to actual system events and entities as recorded in the audit logs. Currently, the calculations treat all nodes in the graph equivalently, even in the use case where the graph nodes represent different entity types. A potential direction for further study is to treat different entities in the multiplex graph as distinct and perform the calculations separately for each node class.

### 5 Results

In this section, we present the results of our attack graph model for three different use cases. The first use case presents a small directed acyclic attack graph (Stergiopoulos et al., 2022). The second use case is a more complex attack graph from (Stergiopoulos et al., 2022) and finally, the third use case is a causal graph that was generated by an attack investigation tool using natural language processing and deep learning (Alsaheel et al., 2021).



### 5.1 Use case 1: Multi-cloud enterprise network

We use a multi-cloud enterprise network as the first use case to demonstrate our modelling method. The network topology and vulnerabilities are adopted from (Stergiopoulos et al., 2022). The first cloud server hosts three virtual machines: a Mail server, a Web server, and a DNS server, and the second cloud server consists of public and private networks. The public network hosts an SQL server and a NAT gateway server, and the private network hosts an Admin server and three virtual machines (VMs). Users outside the network can access the Web server, and employees within the same LAN can access the SQL server through their workstations.

Figure 1 shows the attack graph on the Multi-cloud Enterprise Network that consists of directed links representing the exploits and nodes representing the states. These kinds of networks are directed acyclic graphs (DAGs), i.e., without cycles (Kordy et al., 2014).

The systems have vulnerabilities that are listed in the CVE vulnerabilities database (Mell et al., 2007). The attacker's goal is to compromise a virtual machine in the private network or the database in the public network by gaining root access. The attacker can take different paths in the attack graph.

In Table 2, we show the impact and exploitability values calculated for vulnerabilities in the attack graph (Figure 1). The third column of this table shows the list of links in the attack graph for each vulnerability. In this context, impacts are associated with the properties of vulnerabilities (or edges), while in Equation 5 and Figure 1, successful cyber attacks affect services (or nodes). Our model treats nodes as representations of network states; therefore, all exploits of vulnerabilities that occur on the same node are assumed to have the same impact. If this does not hold, the construction of the attack graph is not consistent, and additional network states (or nodes) should be added to the attack graph.

In Figure 2, we illustrate the non-cumulative impact values of successful cyber attacks from the start node one to other nodes in the graph. For example, the node 7 has a lower impact than nodes 6, 14, and 17 because the vulnerability CVE-2007-4752 from the start node to node 7 has a relatively low impact value. On the other hand, node 10 has a high impact value because the vulnerability CVE-2008-0015, which affects nodes 7 and 12 and is propagated to node 10. As a numerical example, the value of the expected non-cumulative impact of node 10 is  $(0.859 + 0.859 - 0.859 \times 0.859) \times 10.0 \approx 0.98 \times 10 = 9.8$ . The corresponding value of the expected cumulative impact is  $3 \times 6.44 + 9.8 = 29.1$ . This example illustrates how the probabilistic model is applied to calculate expected impacts, and the distinction between non-cumulative and cumulative impacts. The effects on other end nodes can be explained similarly according to Equation 5. In this numerical example and Figure 2, we have used a scaling factor of  $w = 1$  to emphasise the main idea. If a scaling factor is also used, then all exploitability values in Table 2 are multiplied by the scaling factor to get the link weights in the attack graph model.

In our model, we use out-centrality and in-centrality metrics to analyse the propagation of attacks from potential start nodes to potential end nodes. In Figure 1, node one is the start node and node 18 is the end node, but there could be other nodes where the attack chain can start or end. For example, let us consider database servers (dbServer) as the attacker's goal, like in (Stergiopoulos et al., 2022). Equations 1 and 2 define out-centrality and in-centrality metrics as expected values over the whole network structure, not just for possible start and end nodes. There are a few reasons for this. Initially, the start and end nodes might change depending on the situation, or this information may not be available. Furthermore, complete metric data provide details about the structure of the attack graph, which can be used to plan mitigation actions against unknown attack scenarios. In addition, it is straightforward to define metrics for a subset of start nodes and a subset of end nodes similarly to Equation 5.

The out-centrality (Equation 1) and in-centrality (Equation 2) values for the 18 nodes of the Multi-cloud network are shown in

Figure 3. The results are shown for the four link weight values  $w = 0.2, 0.5, 0.8$ , and 1. As the link weights are interpreted as probabilities of successfully exploiting vulnerabilities, lower link weight values correspond to reduced probabilities.

The out-centrality and in-centrality values depicted in Figure 3 illustrate the typical characteristics of directed acyclic networks. Node 1 (Start) has the highest out-centrality, while node 18 (End) has the highest in-centrality. The out-centrality values tend to be more varied than the in-centrality values, as structures near the source nodes and high link weights boost propagation more than structures near target nodes. We also observe that the number of potential alternative paths affects the out-centrality and in-centrality values. Nodes 7, 12, 14, and 15 have high out-centrality values because paths originating from these nodes have many alternatives. Similarly, multiple alternative paths leading to an end node increase its in-centrality value.

When planning, the mitigation actions are optimised based on detailed network modelling and the order of actions can change depending on the exploitability or link weight values of vulnerabilities. Additionally, skilled attackers may utilise longer attack path lengths because their probability of success remains high even with multiple consecutive exploits.

## 5.2 Use case 2: Netflix OSS architecture

The second use case is a combination of containers provided by Netflix, consisting of the Spring cloud ecosystem (Stergiopoulos et al., 2022). Table 3 provides descriptions of the 21 nodes in the NetflixOSS attack graph, the topology of which is depicted in Figure 4.

In Figure 5, we show the out-centrality and in-centrality values of the NetflixOSS attack graph (Stergiopoulos et al., 2022) for four link weight values, and they are found to increase monotonically as the link weight increases. However, the growth rate is slow for low link weights, then increases, and finally slows down again with high link weights due to saturation effects. CVSS scoring (Mell et al., 2007) values are not used in this use case as they have almost equal values in this scenario.

The out-centrality and in-centrality in Figure 5 illustrate the two aspects of the NetflixOSS attack graph. The out-centrality values for the 21 nodes of the graph indicate that nodes 3, 4, 7, 10, 11, 20, and 21 are positioned at the beginning of possible attack chains, while nodes 1, 2, 5, 6, 8, 9, 18, and 19 are at the end of the attack chains. The node seven is identified as the start node of the graph as it has a hundred per cent out-centrality value for the link weight values  $w = 1$ . Nodes 12, 13, 14, 15, 16, and 17 are in the intermediary positions within the attack chains, which can be used to prioritise protective actions or mitigate the impacts of the ongoing cyber attacks.

In Figure 6, the average percentage of exploited nodes is shown when a node's vulnerabilities are mitigated. The results are displayed for link weight values  $w = 0.2$ ,  $w = 0.5$ , and  $w = 0.8$  when the start node is 7. Here the effects are computed as average effects on other nodes. The dotted lines illustrate the average percentage of exploited nodes when the nodes are not protected. The relationship between the link weights of the nodes and the average effect on the network is not linear. For instance, the link weight of  $w = 0.2$  results in only

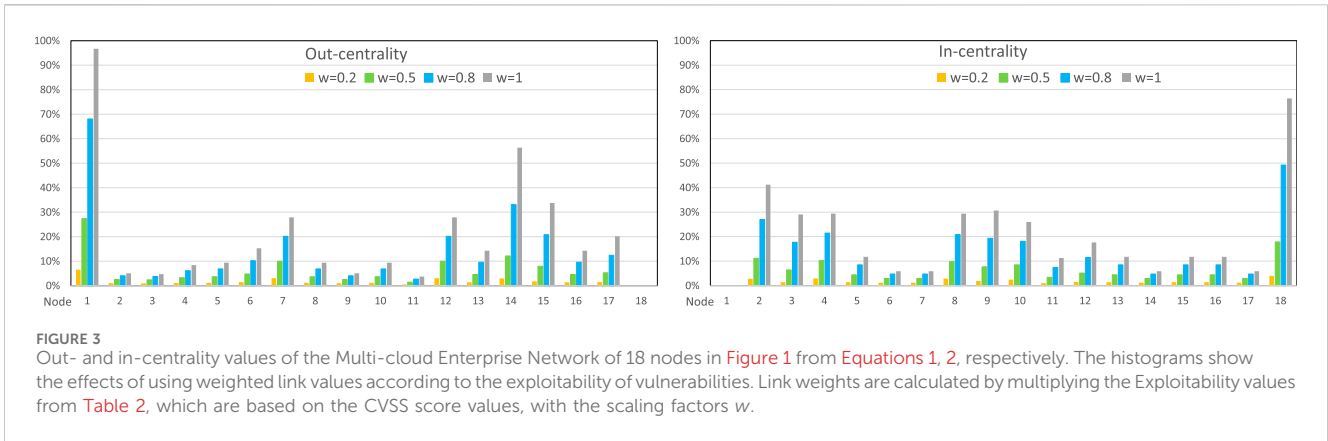


TABLE 3 List of nodes in the NetflixOSS attack graph in Figure 4.

Node	Description	Node	Description
1	Config service (Admin priv)	12	Service b (Admin priv)
2	Config service (User priv)	13	Service b (User priv)
3	Eureka (Admin priv)	14	Service c (Admin priv)
4	Eureka (User priv)	15	Service c (User priv)
5	Hystrix dashboard (Admin priv)	16	Spring cloud dasboard (Admin priv)
6	Hystrix dashboard (User priv)	17	Spring cloud dasboard (User priv)
7	Outside (Admin priv)	18	Turbine (Admin priv)
8	rabbitmq (Admin priv)	19	Turbine (User priv)
9	rabbitmq (User priv)	20	Zuul (Admin priv)
10	Service (Admin priv)	21	Zuul (User priv)
11	Service (User priv)		

about 10 per cent exploitation in the network. Meanwhile, the link weights of  $w = 0.5$  and  $w = 0.8$  lead to approximately sixty and ninety per cent exploitation levels, respectively.

A significant decrease in the number of exploited nodes is seen when the vulnerabilities of the nodes 3, 4, 20, and 21 in the graph are mitigated. This is expected because these nodes are at the beginning of potential attack chains starting from node 7. The impact is bigger when the links have higher weights. Nodes that come right after the start node become more important when the link weight values (exploitability) increase.

When it comes to addressing vulnerabilities, it may be more useful to focus on eliminating vulnerabilities related to services rather than just protecting individual nodes in the attack graph. Next, we compare how out-centrality and in-centrality values decrease when vulnerabilities in services are protected. The comparison will be performed by taking the difference between the unprotected and protected results. Next, we calculate out-centrality (Equation 1) and in-centrality (Equation 2) results for all nodes in the attack graph in Figure 4. In these calculations, we assume that the cyber attack starts at node  $s$  in Equation 1 or nodes  $t$  in Equation 2 with certainty. If the attack starts from the first node in

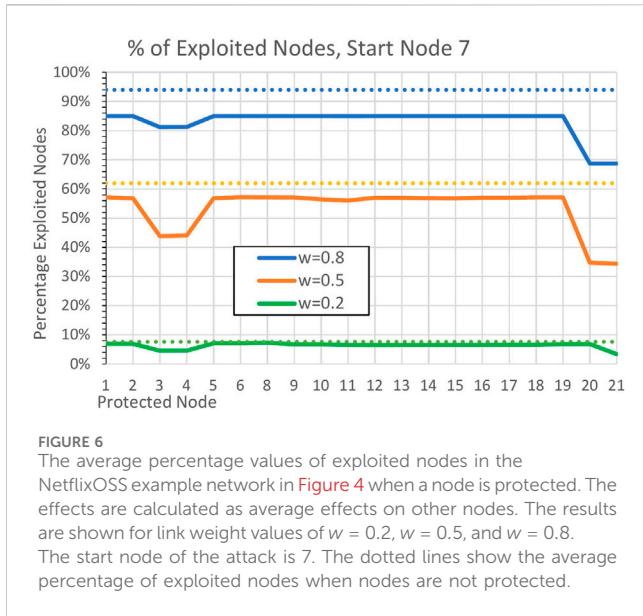
the attack graph in Figure 4, then the relevant results focus solely on node 7.

Figure 7 shows these differences (i.e., decrease) for nodes 1 – 21 after mitigating all the vulnerabilities in the services indicated in the legend. For this example, the link weight is  $w = 0.8$  to represent a skilled perpetrator. Looking at the out-centrality histogram in Figure 7, we observe that 4 bars stand out: Zuul and Eureka services have decreased by 95% and 85% for the out-centrality of node 7, and Eureka services also show a significant decrease for the out-centrality values of node 20 (by 85%) and for node 21 (by 88%).

The in-centrality values shown in Figure 7 indicate that protecting services at the nodes leads to a decrease in the in-centrality values of these particular nodes. For example, Turbine services have a significant impact on nodes 18 and 19. Furthermore, protecting Turbine services results in a substantial decrease in the in-centrality values of nodes 5 and 6. These effects result from nodes 18 and 19 denoting Turbine services states (see Table 3), while nodes 5 and 6 represent subsequent Hystrix dashboard states in the attack graph (Figure 4). On the other hand, protecting services at the beginning of attack chains, such as Zuul and Eureka services, affects







node level indicate that CVE-2017-7376 significantly affects the in-centralities of nodes 1, 2, 4, 11, 13, 17, and 21, with smaller effects on the other nodes. On the other hand, the vulnerability CVE-2017-13,090 has a major impact on nodes 14 and 18, but with minor effects on the other nodes.

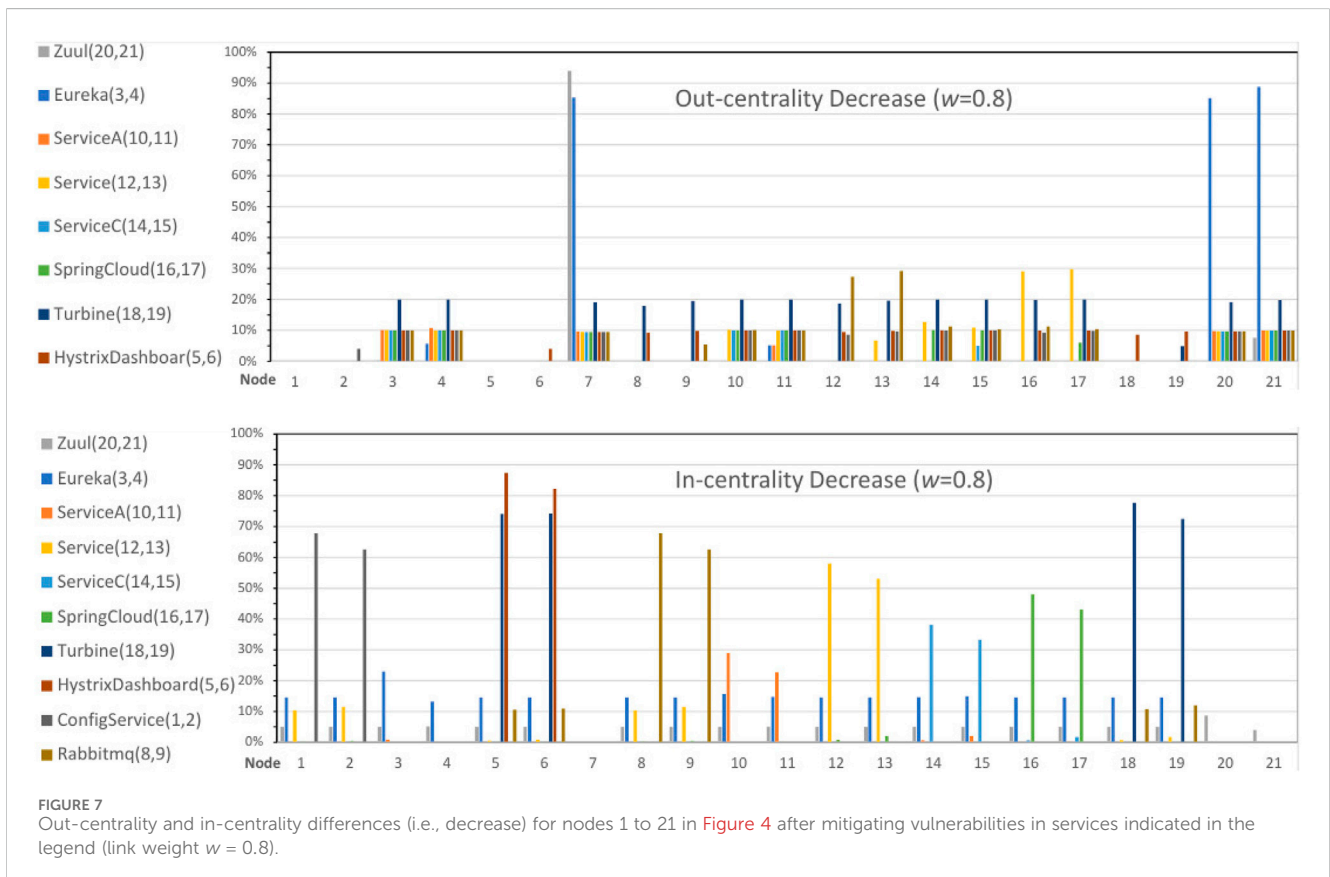
Figure 8b illustrates the decrease in the in-centrality values from the start node seven to other nodes in the graph for the link weight of

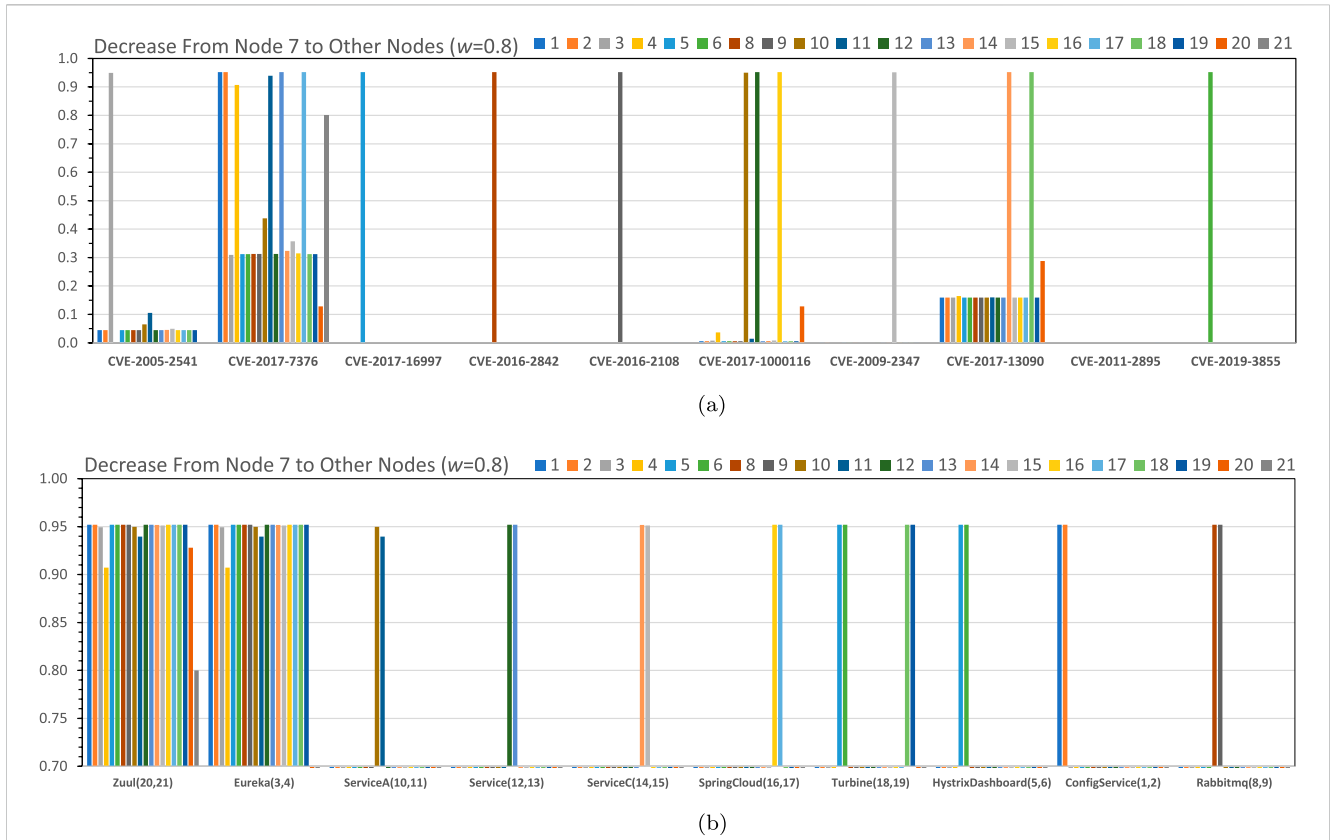
$w = 0.8$  after mitigating vulnerabilities in the services indicated on the horizontal axis. Zuul and Eureka are the most effective services to be protected, as they are at the beginning of attack chains. In Figure 9a we show the summary of the decrease in the average out-centrality from the start node seven to other nodes for the link weight  $w = 0.8$  after mitigating the vulnerabilities. The four most important vulnerabilities in this order are CVE-2017-7376, CVE-2017-13,090, CVE-2017-1000116, and CVE-2005-2541. Figure 9b shows the corresponding results for the services. The most important services are Zuul and Eureka as we have already seen in Figure 8b.

### 5.3 Use case 3: Pony APT campaign

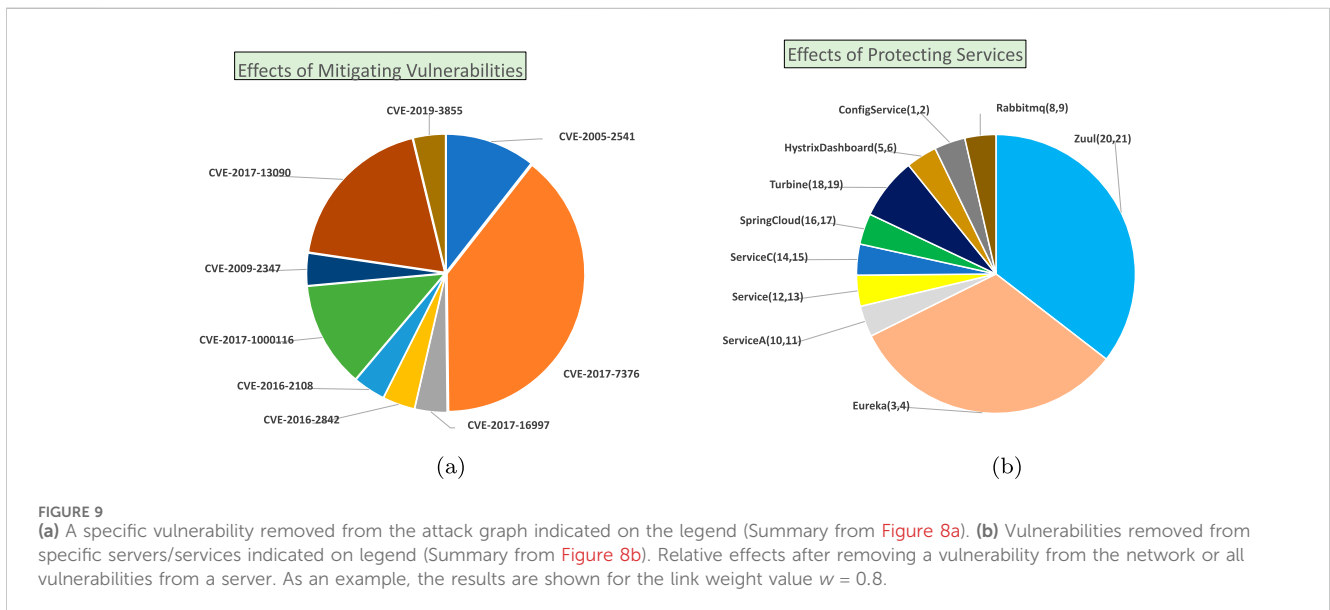
The third use case involves a causal graph generated in (Alsaheel et al., 2021) using an attack investigation tool. This tool integrates natural language processing and deep learning techniques into data analysis to model sequence-based attack and non-attack entities.

Figure 10 depicts attack and non-attack entities detected by the ATLAS investigation tool (Alsaheel et al., 2021). The original attack entities are indicated by the red colour and the original non-attack entities by the green colour (Alsaheel et al., 2021). The blue line shows the extended set of attack nodes reachable from start nodes eight or nine by following all alternative paths in the graph. The red line shows the minimum set of nodes restricted to the original set of attack nodes.



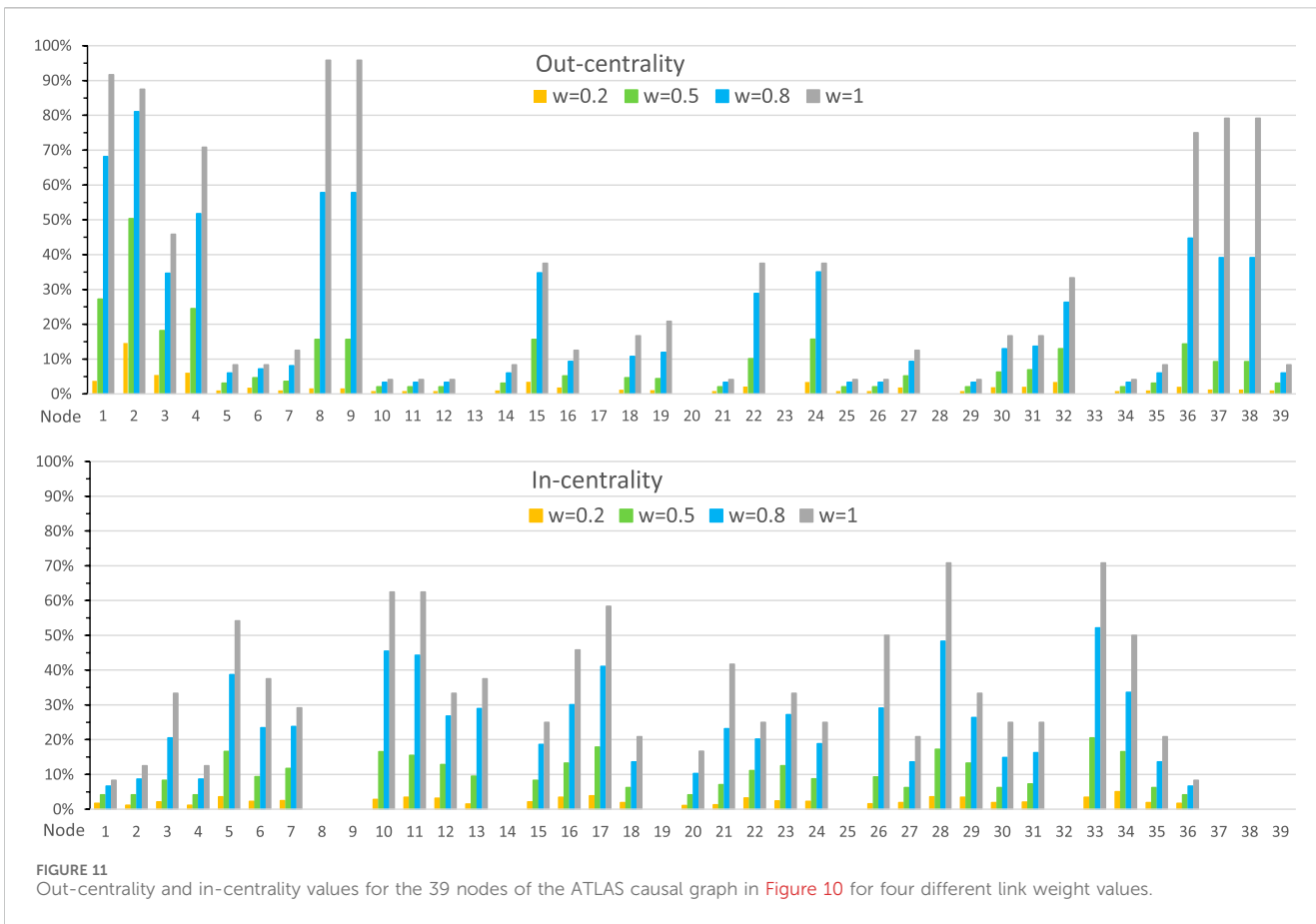


**FIGURE 8** (a) Probability differences (decrease) of the in-centralities from the attack start node seven to other nodes (link weight  $w = 0.8$ ) after mitigating a vulnerability from the network. Vulnerabilities are indicated on the horizontal axis. (b) Probability differences (decrease) of the in-centralities from the attack start node seven to other nodes (link weight  $w = 0.8$ ) after removing all vulnerabilities from a server. Servers are indicated on the horizontal axis. Summaries generated from the model (a) when the entire network is protected against a single vulnerability or (b) a server is protected against all its vulnerabilities.



**FIGURE 9** (a) A specific vulnerability removed from the attack graph indicated on the legend (Summary from Figure 8a). (b) Vulnerabilities removed from specific servers/services indicated on legend (Summary from Figure 8b). Relative effects after removing a vulnerability from the network or all vulnerabilities from a server. As an example, the results are shown for the link weight value  $w = 0.8$ .





exploited. This outcome is attributed to the critical positions of these nodes within the graph structure. In the table of Figure 10, the tool’s classified attack nodes are indicated in red, non-attack nodes in blue, and non-reachable nodes (both attack and non-attack) in black. When prioritising mitigation measures, nodes 22 and 15 have significant effects on the extended graph, although they have not been recognised as attack nodes by the investigative tool.

In contrast to the networks in use cases 1 and 2, the ATLAS causal graph (Figure 10) has a loop between the nodes 15, 22, and 24, and bidirectional links between the nodes 10 and 11, and between the nodes 30 and 31. In our model, we assume that due to a loop and bi-directionality, circular and recurrent effects are possible. Figure 13 illustrates the circular effects in the ATLAS causal graph for three different link weight values. The histogram shows a difference between two situations, namely, one with circular effects and the other with only self-avoiding paths in the network structure. The effects vary according to the weight values of the links. They are low when the link weights are  $w = 0.2$ , peak at higher link weights, and gradually slow down due to saturation effects for high link weight values.

To summarise, it turns out that for the attack graph with one loop and two bidirectional links, the out-centrality values are increased. Furthermore, nodes 1, 2, 8, and 9 have increasing out-centrality values because they are at the beginning of attack paths that can reach the loop nodes 15, 22 or 24. Upon examining the graph in Figure 12, we conclude that the in-centrality values of

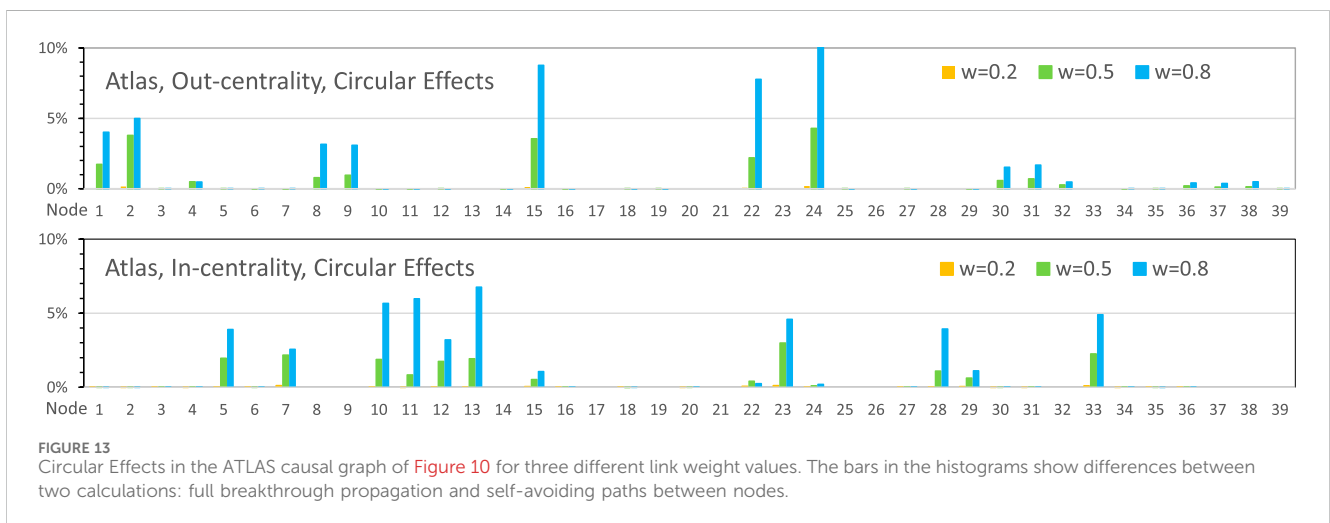
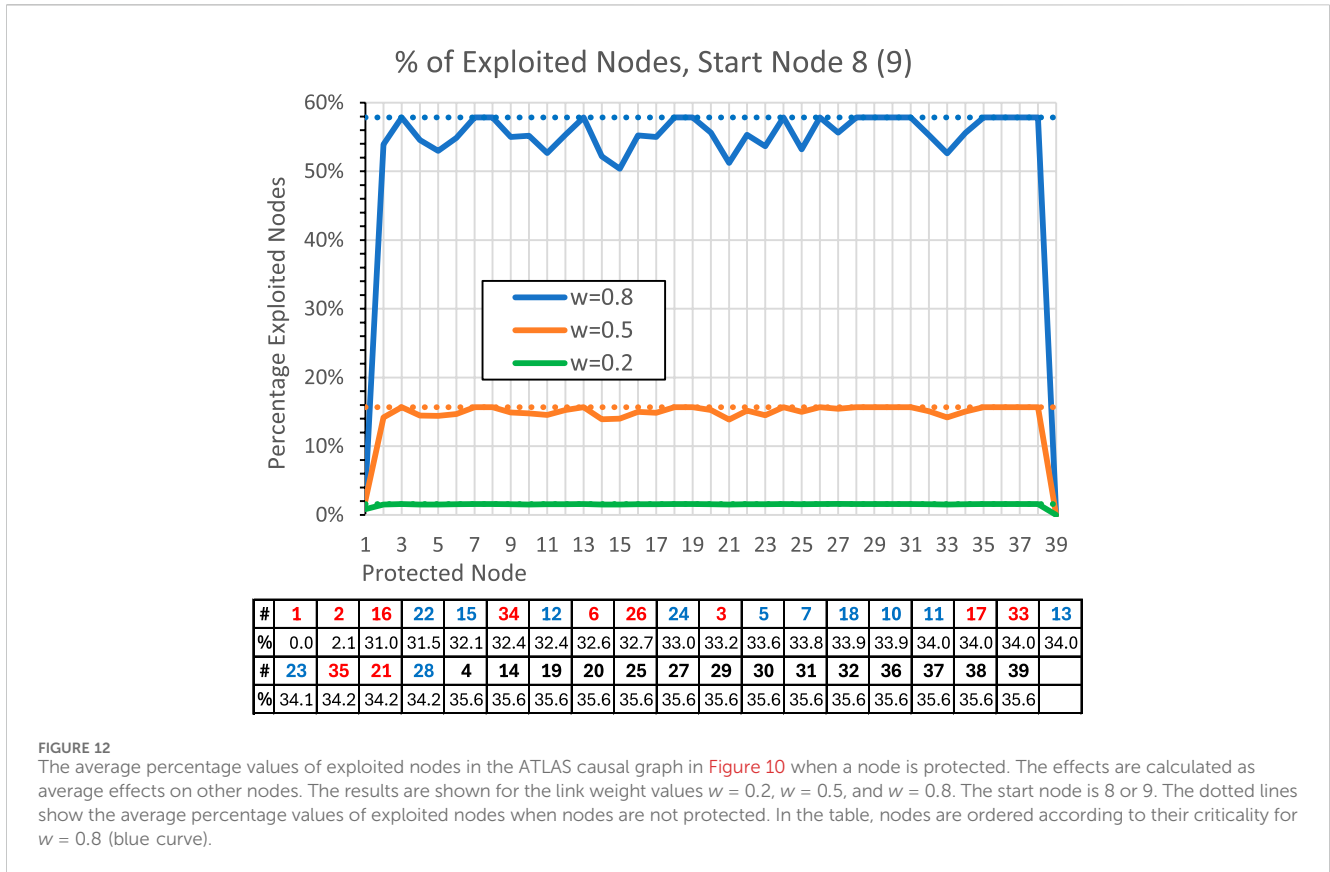
nodes 5, 7, 10, 11, 12, 13, and 23 have increased due to the circular effects of the loop, and the in-centrality values of nodes 28, 29 and 33 have increased due to the recurrent effects between nodes 30 and 31 by the two bidirectional links.

## 6 Concluding remarks

In cybersecurity analysis of computer networks and service systems, it is natural to take a graph-based approach. Until recently, cyber-related graphs, such as attack graphs, have been created manually or generated by scanning network configurations to identify vulnerabilities and compute potential attack paths. In this case, the analysis of an attack graph involves examining individual paths that an attacker could take and evaluating the risk associated with each path.

In the present study, we have introduced a novel probabilistic graph-based analysis approach to investigate the structural and dynamic properties of the attack and causal graphs. This approach has previously been used in our research to model the spread of influence in complex networks. Here, we are applying this methodology for the first time to analyse the propagation of attacks and causal influences. In this context, out-centrality and in-centrality measures are used to identify critical nodes in the graph. The documented exploitability values of vulnerabilities mimic the probability of propagation of attacks or causal influences on possible paths or sequences of actions that an





attacker or influencer can follow to achieve a particular objective, such as compromising a critical asset in a network. Similarly, the documented impact values of vulnerabilities are used to demonstrate the model in analysing cumulative and individual impacts of attacks.

Future research could focus on multilayer modelling of enterprise infrastructure and attack graphs. Additional areas of interest could include the integration of real-time attack graph generation and the exploration of hybrid approaches that combine machine learning or anomaly detection methods. Recently, there has emerged a variety of studies on AI-driven

techniques in cybersecurity (Salem et al., 2024) and intrusion detection (Por et al., 2024). These approaches offer new opportunities to be integrated with the probabilistic methods discussed in this study.

In summary, we have developed a new method for analysing alternative paths within an attack graph and calculating their combined exploitability and impact. This approach differs from the previous methods since our approach offers a probabilistic framework to assess the combined effects of the entire attack graph, rather than focusing solely on the maximum or minimum

scenarios. We have applied our model by presenting results related to functional services based on identified vulnerabilities. Additionally, this method can be extended to analyse multiple attack graphs simultaneously in a network structure. For higher levels of abstraction and summaries, it is essential to consistently aggregate alternative attack or causal effect paths within the model. This integrated perspective improves understanding of the situation and helps prioritise mitigation efforts. Thus, our model could be a versatile tool in the hands of cyber analysts.

## Data availability statement

The original contributions presented in the study are included in the article/[Supplementary Material](#), further inquiries can be directed to the corresponding author.

## Author contributions

VK: Methodology, Visualization, Conceptualization, Validation, Writing – original draft, Writing – review and editing, Formal Analysis, Software. LP: Writing – review and editing, Writing – original draft. TT: Project administration, Writing – review and editing, Writing – original draft. KK: Funding acquisition, Writing – original draft, Supervision, Writing – review and editing.

## Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

## References

- Aksu, M. U., Bicakci, K., Dilek, M. H., Ozbayoglu, A. M., and Tatli, E. I. (2018). Automated generation of attack graphs using NVD. *Proc. Eighth ACM Conf. Data Appl. Secur. Priv.*, 135–142. doi:10.1145/3176258.3176339
- Almiala, I., Aalto, H., and Kuikka, V. (2023). Influence spreading model for partial breakthrough effects on complex networks. *Phys. A Stat. Mech. its Appl.* 630, 129244. doi:10.1016/j.physa.2023.129244
- Alsaheel, A., Nan, Y., Ma, S., Yu, L., Walkup, G., Celik, Z. B., et al. (2021). “[Atlas]: a sequence-based learning approach for attack investigation,” in *30th USENIX security symposium (USENIX security 21)*, 3005–3022. Available online at: <https://www.usenix.org/conference/usenixsecurity21/presentation/alsaheel>.
- Alserhani, F. M. (2015). Knowledge-based model to represent security information and reason about multi-stage attacks. *Lect. Notes Bus. Inf. Process.* 27, 482–494. doi:10.1007/978-3-319-19243-7\_44
- Carter, K. M., Idika, N., and Streilein, W. W. (2014). Probabilistic threat propagation for network security. *IEEE Trans. Inf. Forensics Secur.* 9 (9), 1394–1405. doi:10.1109/TIFS.2014.2334272
- FIRST (2025). Common vulnerability scoring system sig (forum of incident response and security teams). Available online at: <https://www.first.org/cvss/>.
- Haque, M. A., Shetty, S., Gold, K., and Krishnappa, B. (2021). “Realizing cyber-physical systems resilience frameworks and security practices,” in *Security in cyber-physical systems: foundations and applications*, 1–37. doi:10.1007/978-3-030-67361-1\_1
- Homer, J., Zhang, S., Ou, X., Schmidt, D., Du, Y., Rajagopalan, S. R., et al. (2013). Aggregating vulnerability metrics in enterprise networks using attack graphs. *J. Comput. Secur.* 21 (4), 561–597. doi:10.3233/JCS-130475
- Hutchins, E. M., Cloppert, M. J., and Amin, R. M. (2011). “Intelligence-driven computer network defense informed by analysis of adversary campaigns and

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

## Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fcpxs.2025.1620260/full#supplementary-material>

intrusion kill chains,” in *Proceedings of the 6th international conference on information warfare and security (ICIW 2011)*. Washington, DC: Academic conferences and publishing international limited. Available online at: <https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf>

Ibrahim, A., Bozhinoski, S., and Pretschner, A. (2019). “Attack graph generation for microservice architecture,” in *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*, 1235–1242. doi:10.1145/3297280.3297401

Jiang, G., Mohandas, R., Leathery, J., Berry, A., and Galang, L. (2017). CVE-2017-0199: in the Wild attacks leveraging HTA handler — mandiant. Available online at: <https://cloud.google.com/blog/topics/threat-intelligence/cve-2017-0199-hta-handler/>.

Kordy, B., Piètre-Cambacédès, L., and Schweitzer, P. (2014). Dag-based attack and defense modeling: don’t miss the forest for the attack trees. *Comput. Sci. Rev.* 13–14, 1–38. doi:10.1016/j.cosrev.2014.07.001

Kuikka, V., and Kaski, K. K. (2024). Detailed-level modelling of influence spreading on complex networks. *Sci. Rep.* 14 (28069), 28069. doi:10.1038/s41598-024-79182-9

Kuikka, V., Aalto, H., Ijäs, M., and Kaski, K. K. (2022). Efficiency of algorithms for computing influence and information spreading on social networks. *Algorithms* 15 (8), 262. doi:10.3390/a15080262

Lallie, H. S., Debattista, K., and Bal, J. (2020). A review of attack graph and attack tree visual syntax in cyber security. *Comput. Sci. Rev.* 35, 100219. doi:10.1016/j.cosrev.2019.100219

Landherr, A., Friedl, B., and Heidemann, J. (2010). A critical review of centrality measures in social networks. *Bus. and Inf. Syst. Eng.* 2 (6), 371–385. doi:10.1007/s12599-010-0127-3

- Levner, E., and Tsadikovich, D. (2024). Fast algorithm for cyber-attack estimation and attack path extraction using attack graphs with and/or nodes. *Algorithms* 17 (11), 504. doi:10.3390/a17110504
- Li, Y., and Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. *Energy Rep.* 7, 8176–8186. doi:10.1016/j.egyr.2021.08.126
- Liu, C., Singhal, A., and Wijesekera, D. (2012). "Using attack graphs in forensic examinations," in *2012 seventh international conference on availability, reliability and security*, 596–603. doi:10.1109/ARES.2012.58
- Mell, P., Scarfone, K., and Romanosky, S. (2007). A complete guide to the common vulnerability scoring system version 2.0. Available online at: [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=51198](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198).
- NIST (2012). Nist sp 800-30: guide for conducting risk assessments (joint task force transformation initiative working group and others). doi:10.6028/NIST.SP.800-30r1
- Noel, S., and Jajodia, S. (2017). "A suite of metrics for network attack graph analytics," in *Network security metrics* (Springer International Publishing), 141–176. doi:10.1007/978-3-319-66505-4\_7
- Pendleton, M., Garcia-Lebron, R., Cho, J.-H., and Xu, S. (2016). A survey on systems security metrics. *ACM Comput. Surv. (CSUR)* 49 (4), 1–35. doi:10.1145/3005714
- Qi, Y., Gu, Z., Li, A., Zhang, X., Shafiq, M., Mei, Y., et al. (2023). Cybersecurity knowledge graph enabled attack chain detection for cyber-physical systems. *Comput. Electr. Eng.* 108, 108660. doi:10.1016/j.compeleceng.2023.108660
- Salem, A. H., Azzam, S. M., Emam, O. E., and Abohany, A. A. (2024). Advancing cybersecurity: a comprehensive review of ai-driven detection techniques. *J. Big Data* 11 (1), 105. doi:10.1186/s40537-024-00957-y
- Segovia-Ferreira, M., Rubio-Hernan, J., Cavalli, A., and Garcia-Alfaro, J. (2024). A survey on cyber-resilience approaches for cyber-physical systems. *ACM Comput. Surv.* 56 (8), 1–37. doi:10.1145/365295
- Shakarian, P., Bhatnagar, A., Aleali, A., Shaabani, E., and Guo, R. (2015). "The independent cascade and linear threshold models," in *Diffusion in social networks* (Springer International Publishing), 35–48. doi:10.1007/978-3-319-23105-1\_4
- Sheyner, O., and Wing, J. (2003). "Tools for generating and analyzing attack graphs," in *International symposium on formal methods for components and objects*, 344–371. doi:10.1007/978-3-540-30101-1\_17
- Sikos, L. F. (2023). Cybersecurity knowledge graphs. *Knowl. Inf. Syst.* 65 (9), 3511–3531. doi:10.1007/s10115-023-01860-3
- Stergopoulos, G., Dedousis, P., and Gritzalis, D. (2022). Automatic analysis of attack graphs for risk mitigation and prioritization on large-scale and complex networks in industry 4.0. *Int. J. Inf. Secur.* 21 (1), 37–59. doi:10.1007/s10207-020-00533-4
- Stojanović, B., Hofer-Schmitz, K., and Kleb, U. (2020). Apt datasets and attack modeling for automated detection methods: a review. *Comput. and Secur.* 92, 101734. doi:10.1016/j.cose.2020.101734
- Wachter, J. (2023). Graph models for cybersecurity – a survey.
- Wang, L., Islam, T., Long, T., Singhal, A., and Jajodia, S. (2008). "An attack graph-based probabilistic security metric," in *Data and applications security XXII: 22nd annual IFIP WG 11.3 working conference on data and applications security London, UK, July 13-16, 2008 proceedings* 22, 283–296. doi:10.1007/978-3-540-70567-3\_22
- Yee Por, L., Dai, Z., Juan Leem, S., Chen, Y., Yang, J., Binbeshr, F., et al. (2024). A systematic literature review on ai-based methods and challenges in detecting zero-day attacks. *IEEE Access* 12, 144150–144163. doi:10.1109/ACCESS.2024.3455410
- Zeng, J., Wu, S., Chen, Y., Zeng, R., and Wu, C. (2019). Survey of attack graph analysis methods from the perspective of data and knowledge processing. *Secur. Commun. Netw.* 2019 (1), 1–16. doi:10.1155/2019/2031063
- Zenitani, K. (2023). Attack graph analysis: an explanatory guide. *Comput. and Secur.* 126, 103081. doi:10.1016/j.cose.2022.103081