RAID: RL-Based Framework for Disrupting Adversarial Information in Battlefields

Muhammad Salman[a], Taehong Lee[b], Ali Hassan[a], Muhammad Yasin[a], Kiran Khurshid[a], Youngtae Noh[c,*]

[a] *College of Electrical & Mechanical Engineering, National University of Sciences and Technology (NUST), Islamabad, Pakistan.*
[b] *Korea Institute of Energy Technology (KENTECH), Naju, South Jeolla, Republic of Korea.*
[c] *Department of Data Science, Hanyang University, Seoul, Republic of Korea*

Abstract

During battlefield operations, military radios (hereafter nodes) exchange information among various units using a mobile ad-hoc network (MANET) due to its infrastructureless and self-healing capabilities. Adversarial cyberwarfare plays a crucial role in modern combat by disrupting communication between critical nodes (i.e., nodes mainly responsible for propagating important information) to gain dominance over the opposing side. However, determining critical nodes within a complex network is an NP-hard problem. This paper formulates a mathematical model to identify important links and their connected nodes, and presents RAID, a reinforcement learning-based framework with an encoder-decoder architecture, for efficiently detecting and jamming critical nodes. The encoder transforms network structures into embedding vectors, and the decoder assigns a score value to the embedding vector with the highest reward. Our framework is trained and tested on custom-built MANET topologies using the Named Data Networking (NDN) protocol. RAID has been evaluated across various scales and weighting methods for both connected node and network dismantling problems. Our proposed method outperformed existing RL-based baselines, with a 24% performance gain for smaller topologies (50-100 nodes) and 8% for larger ones (400-500 nodes) in connected node problems, and a 7% gain for smaller topologies and 15% for larger ones in network dismantling problems.

Keywords: Mobile Adhoc Network (MANET), RL-based Adversarial Information Disruptor (RAID), Named Data Networking (NDN), Graph Embedding, Battlefield.

1. Introduction

On the battlefield, each side strives to secure dominance over the other. One side achieves this by exchanging real-time information through resilient communication networks and state-of-the-art communication devices. For communication networks, the military uses mobile ad-hoc wireless networks (MANETs) due to their infrastructure-less (i.e., do not require infrastructure) and self-healing capabilities [1; 2]. Similarly, to exchange information over MANETs, they utilize state-of-the-art military radio devices [3] and the Internet of Battlefield Things (IoBT – including wearable devices and sensors related to the battlefield) [4; 5]. Reliable communication through a resilient network empowers them to develop real-time strategies that significantly enhance the effectiveness and precision of their operations. Conversely, the opposing side employs sophisticated tactics to undermine these communication efforts. They use cyberwarfare attacks[1] to disrupt the communication occurring between military entities [6], thereby enhancing their dominance over the adversary [7].

In modern battlefields, cyberwarfare stands out as a key factor in determining dominance. Several nations, including France, Japan, and Germany, are actively incorporating cyberwarfare into their military doctrines [8]. The objective in cyberwarfare is to identify and disrupt the communication of adversary military nodes that are conveying important information. These nodes, known as critical nodes, are determined by analyzing the communication patterns. Critical nodes are frequently engaged in communication with their neighbors and are linked with more military nodes in the surrounding area compared to normal military nodes, making them more socially connected. Once identified, a jamming strategy is applied to disrupt the communication occurring in these critical links.

Unfortunately, identifying and jamming the critical nodes on the battlefield is not straightforward due to the highly complex and dynamic nature of the network environment. This complexity arises from two main factors. First, the military nodes (specifically RF nodes possessed by military personnel) using the infrastructure-less network are highly mobile. Second, the scale of nodes in a given area is unpredictable [9]. In such kind of situation, detecting a critical node itself is an NP-hard problem [10]. In this paper, we address this problem by developing a Reinforcement Learning-based framework known as RAID[2]

*Y. Noh is the corresponding author

Email addresses: msalman@ceme.nust.edu.pk (Muhammad Salman), etehong@kentech.ac.kr (Taehong Lee), alihassan@ceme.nust.edu.pk (Ali Hassan), m.yasin@ceme.nust.edu.pk (Muhammad Yasin), kiran.khurshid@ceme.nust.edu.pk (Kiran Khurshid), youngtaenoh@hanyang.ac.kr (Youngtae Noh)

[1] In the context of the battlefield, cyberwarfare involves using digital attacks to disrupt, degrade, or disable the enemy's communication networks, information systems, and critical infrastructure, thereby weakening their ability to coordinate, communicate, and execute military operations effectively.

[2] RAID stands for RL-based Adversarial Information Disruptor

to effectively detect and disrupt critical nodes in complex networks of varying scales. We assess the impact of removing these nodes on overall network connectivity through two key problems: (1) the connected node problem, which examines how different node removal strategies affect network connectivity; and (2) the network dismantling problem, which aims to fragment the network into smaller components with minimal node removal, measuring the network's resilience by observing connectivity degradation. The detailed contributions of this work are summarized as follows:

- We have derived a comprehensive mathematical model for determining critical nodes in a complex network that are mainly responsible for information spreading. Based on this derivation, we formulated our problem statement for jamming the identified critical nodes by considering both the connected node and the network dismantling problem.

- We developed a Reinforcement Learning-based framework incorporating an encoder-decoder architecture. The encoder leverages a Graph Neural Network (GNN) to transform complex network information into node embeddings by iteratively aggregating the feature weights of the node and its neighborhood. The decoder processes these node embeddings using a scoring function (Q-score) to assign a value reflecting each node's importance within the network. Based on the highest Q-score, a greedy exploration rate (ϵ) is used to remove the corresponding (critical) node.

- We conducted an extensive evaluation using MANET topologies that we created by applying adaptations to the ndnSIM [11; 12] in the ns-3 network simulator. We assessed the reliability of our framework by examining the detection of randomly selected critical nodes' transmissions over short durations in complex networks. Moreover, we evaluated the framework's scalability and performed a comprehensive analysis of the connected node and network dismantling problems across various scales and weighting methods.

2. Background

Military operations have unique features on the battlefield. These include a clear objective that prioritizes the mission over individual interests. They maintain a strict hierarchical structure with consistent information and command exchange [13]. Their goal is to carefully plan strategies, considering factors like buildings, terrain, distances, etc., to gain dominance over the opposing side. The crucial element in disseminating commands and information to every military unit depends on the infrastructure of military communication. In this section, we will discuss the evaluation of military communication and its transition toward the modern military communication, the hierarchical structure of the modern military, and their mobility distribution.

2.1. Evolution in Military Communication

Military communication has undergone significant evolution throughout the centuries, adapting to advancements in technology, strategic requirements, and the dynamic nature of modern warfare. In the Pre-Electric Era (around two centuries ago), communication was typically confined to the range of visibility or the pace at which an individual could move. They employed a variety of methods for message communication during wars, utilizing means such as wagons, horseback, and foot to convey crucial information. For instance, in 490 BC, a Greek courier covered a distance of twenty-six miles to deliver news of a military victory to Athens [14].

The revolution came in military communication after the advention of electricity. The first significant revolution in military signaling occurred in the mid-nineteenth century with the introduction of electric telegraphy [15]. This innovation allowed messages to be transmitted to stations hundreds of miles apart within a matter of minutes. After two decades, Alexander Graham Bell invented the telephone, and it was subsequently utilized by Britain for specialized purposes in the military [16]. The primary drawback of using the telephone during the war was its intricate infrastructure and the cumbersome nature of the wires.

After a half-century, from 1895 to 1914, the second revolution in communications unfolded with the advent of wireless telegraphy for military communication [17]. Both wireless and wired communication underwent significant testing and utilization during the First World War. The military radio was also tested during this war. Military wireless communication was further revolutionized in the Second World War with the invention of more sophisticated radio systems (e.g., frequency modulation (FM) for local communication in sea and lands, walkie-talkie transceivers etc). In addition, during this period, a cyberwarfare equipment was introduced, enabling the hacking (or code-breaking) and retrieval of crucial information from the radio frequency communication links of opposing forces [18]. The US in the 1960s also introduced its communication standard called the World Wide Military Command and Control System (WWMCCS) for integrating all their defense communication systems [19].

The third and fourth revolutions brought about comprehensive transformations in the military communication structure. In the third revolution, the invention of transistors at Bell Labs [20] resulted in the substitution of fragile vacuum tubes with transistors in radio communication equipment. Additionally, this era witnessed significant advancements in the encryption of sensitive messages during wartime [21]. The fourth revolution, which unfolded in the mid-seventies, saw the digitization of military communication, accompanied by the introduction of communication satellites for battlefield communication. These satellites played a crucial role in providing GPS information, proving particularly invaluable in desert areas like the Gulf Wars [22]. To date, advanced military radios, the Internet of Battlefield Things (IoBT), and infrastructure-free, self-healing networks are employed for information exchange and communication in warfare.

2.2. Network infrastructure of Military Communication

The tactical region presents numerous challenges for various reasons. First, the battlefield can be unpredictable, featuring diverse elements such as buildings and varied terrain. Second, the presence of diverse platforms with numerous tools and devices operating on the same communication channel contributes to substantial channel impairments. Lastly, the battlefield is inhabited by military personnel engaged in diverse forms of communication, including interactions with naval ships, the airforce, and military vehicles, exacerbating interference in tactical networks [23]. It is important to note that the structured network infrastructure (e.g., 5G network) during the battlefield is infeasible to use owing to to the risk of physical damage from combat activities and the potential for hacking, which could compromise sensitive information. In response to these challenges, the Mobile Adhoc Network (MANET) emerges as a specialized tactical-purpose network infrastructure capable of addressing the above mentioned issues. MANET is a structureless and self-organized network on demand and can be created when desired [23]. It supports dynamic topologies, making it best suited for the changing conditions on the battlefield; and is highly scalable. Its decentralized nature contributes to resilience, and when complemented with cryptographic measures, it has the potential to achieve a high level of security [24]. MANET also supports the tactical mobility of military units. Due to these features, MANET is gaining significant popularity in modern tactical wars.

2.3. Military Mobility Models

The military units have unpredictable and random mobility patterns. The front-line fighters are infantry units, comprised of soldiers who exhibit a random walk (RW) model [25]. In conjunction with infantry units, military vehicles, including tanks and artillery vehicles equipped with communication devices, are integral to the operational setup. These military vehicles operate on a random waypoint model (RWM) and establish communication with infantry units to offer assistance as needed. Simultaneously, they communicate with the operation center, which has zero mobility, to request aerial or marine support. The detailed illustration is shown in Figure 1 with the mobility models along with their communication links. These elements have distinct communication roles during wartime. As an example, if the soldiers in the infantry unit decide to alter their ground strategy, they can communicate this change to their military vehicles. Subsequently, the military vehicles relay the message to the operation center, enabling the implementation of further strategic actions such as launching attacks from the air, sea, or modifying their maneuvers.

In the following section, we elaborated on the mathematical model, focusing on the dissemination of information among military entities. We also explored how we calculated the importance of communication links between these entities and discussed methods for identifying critical nodes that could potentially pose attack threats.

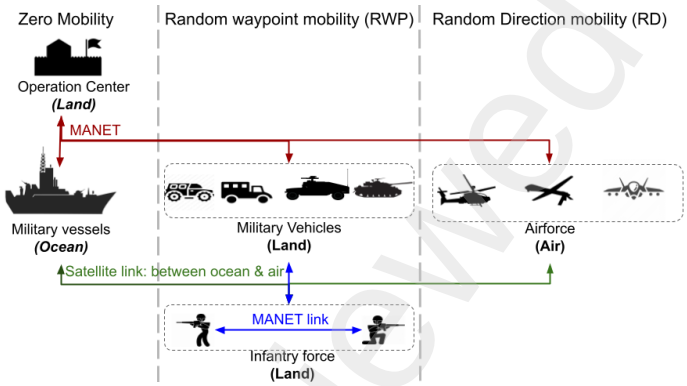


Figure 1: Exchange of communication between different hierarchical groups in the battlefield.

3. Mathematical Model

In this section, we articulate the mathematical model for identifying the most critical nodes. These node are adversarial military entities for disseminating (or spreading) information that may lead to a potential attack. Initially, we focus on determining the significance of each communication link among adversarial military entities [26]. We then create a mathematical model to identify the critical nodes connected through these links that are responsible for information dissemination in a complex battlefield network.

Our problem bears a resemblance to the susceptible-infected-susceptible (SIS) model of epidemic spreading [27]. We have a total of N nodes representing military entities and L edges denoting the links between them, characterized by the adjacency matrix A. Each node i can exist in one of two states, denoted as σ_i: it can either be in a state of potentially receiving the information (i.e., susceptible – S) or it has already received the information from the sender military entity (i.e., infected – I). Therefore $\sigma_i \in \{S,I\}$. This implies that the subsequent military entity i is prone to receiving information from the one who has previously received it, i.e., j. Specifically, the $\{i, j\}$ link is in an SI state if σ_i is susceptible and σ_j is infected. The state of the communication link between two military entities is further characterized by two parameters: β and μ; where, β represents the probability of successful transmission (or infection), while μ denotes the probability of transitioning from the transmission state to its non-transmission state (recovery).

3.1. Information Spreading Among Connected Components

Our objective is to mitigate the spread of information among hierarchical groups within the military, as such dissemination could potentially result in a dominant strategic move by the opposing side. To achieve this, we employ bond percolation, aiming to identify and, consequently, remove the critical node primarily responsible for information spread. One method to tackle this is by setting a threshold for information spread within a spectral radius, as suggested in [26]. However, a major drawback here is the information continues to spread until it reaches the specified threshold. Moreover, this approach may be better suited for scenarios with a deterministic network

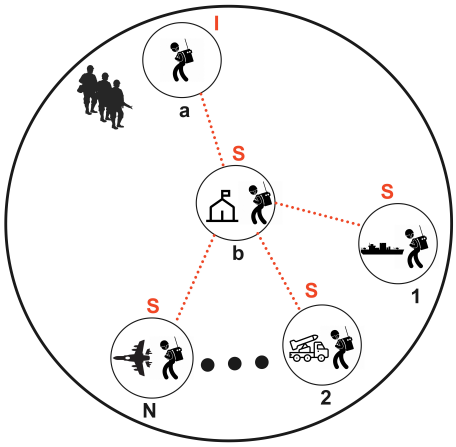


Figure 2: Information spreading between military entities.

topology, unlike the dynamic and unpredictable nature of the battlefield topology. To address this more effectively, we implement an information containment strategy at both node and link levels. In particular, we assess the nodes' and links' probabilities during the information exchange. For instance, $P(\sigma_i = S)$ represents the probability that node i is likely to receive information from the other military entity, or $P(\sigma_i = S, \sigma_j = I)$ represents the *SI* state probability of the {*i,j*} link that was used for the information exchange.

Let us consider a simplified example depicted in Figure 2. In this illustration, the military entity a faces a threat from opposing forces. This node aims to communicate the current battlefield situation (i.e., information about the impending danger)[3], through a MANET link to its upper hierarchy (e.g., headquarter) denoted by b. In this scenario, a is considered to be in an infectious state "*I*", and the link {a, b} is in an *IS* state. These pieces of information can be successfully transmitted from a to b with a probability β. The node b (i.e., headquarter) is also linked with other supporting entities, such as air raid services, artillery, and marine forces, capable of providing assistance to a based on their demands. As a result, all the other entities {*1, 2, .. N*} are likely to receive aid-related information from the headquarters b, placing them in a susceptible state *S*. The probability of information spreads in the nodes can be expressed as:

$$\bar{\eta}_{ab} = \beta P(\sigma_b = S, \sigma_a = I) \sum_{n=1}^{N} A_{bn} \beta P(\sigma_n = S \mid \sigma_b = I) \quad (1)$$

where, $P(\sigma_b = S, \sigma_a = I)$ illustrates the joint probability of how the information is likely to be transported to node b from a through the link {a, b} with the probability of successful transmission denoted as β. $P(\sigma_n = S \mid \sigma_b = I)$ is the conditional probability, which shows how all the associated N nodes are likely to receive the information if b bears it. To prevent the transfer of information from a to b, it is necessary to remove the link {a, b}. It is important to note that breaking the communication between the link {a, b} might be useful if b is not infected.

[3]The goal is to implement a counter-strategic move

On the contrary, once the information has already propagated to b and it has entered an infected state *I*, the removal of the link {a, b} will not have a significant impact. We can further approximate that the military entity b, along with all associated nodes (i.e., {*1, 2, ..., N*}), are receptive to the information originating from the military entity a in the following manner:

$$\bar{\eta}_{ab} \approx \beta P(\sigma_b = S) P(\sigma_a = I) \sum_{n=1}^{N} A_{bn} \beta P(\sigma_n = S) \quad (2)$$

It is noteworthy that the information exchange can occur in either half-duplex or full-duplex mode. For instance, if node b is aware of the impending threat to a, then node b can reciprocate by sharing an updated strategic move with a. In such cases, the information flow will be in the reverse direction. Therefore, we can denote it as follows:

$$l_{ab} = \bar{\eta}_{ab} + \bar{\eta}_{ba} \quad (3)$$

where l_{ab} is link importance, which denotes the information propagation in a forward way "$\bar{\eta}_{ab}$", as well as in a reverse way "$\bar{\eta}_{ba}$".

3.2. Link Importance Within and Between Subnets

In a battlefield setting, the topology is more complex, and communication occurs within and between subnets[4], as illustrated in Figure 3. Here, we consider the presence of two subnets, each with an average interconnection among military nodes denoted as $\langle k \rangle$. More precisely, $\langle k \rangle_A$ signifies the average degree of connected nodes in subnet A, while $\langle k \rangle_B$ represents that in subnet B. Moreover, we make the assumption that the number of connected nodes within the military nodes in both subnets are not equal, ensuring a random distribution of nodes within each subnet, i.e., $\langle k \rangle_B > \langle k \rangle_A$. Let us also assume that the successful communication exchange (or information dissemination among the military entities) is denoted by ρ (i.e., ρ^A for subnet A and ρ^B for subnet B). Therefore, we can determine the link importance within and between the subnets by using homogeneous mean-field approximation [28]. To do so, we substitute $P(\sigma_a = I) \approx \rho$ and $P(\sigma_b = S) \approx 1 - \rho$. The link importance within a subnet can be denoted by the following expressions:

$$l_A \approx 2\beta^2 \rho^A (1 - \rho^A)^2 \langle k \rangle_A \quad (4)$$

$$l_B \approx 2\beta^2 \rho^B (1 - \rho^B)^2 \langle k \rangle_B \quad (5)$$

Similarly, the link importance between the subnets A and B can be expressed as:

$$l_{AB} \approx \beta^2 \left[\rho^A (1 - \rho^B)^2 \langle k \rangle_B + \rho^B (1 - \rho^A)^2 \langle k \rangle_A \right] \quad (6)$$

In Equations (4) ~ (6) above, two parameters are crucial in estimating the importance of a link within and between subnets. The first parameter, denoted as $\langle k \rangle$ reflects the average degree of

[4]Within means, the communication within the military entities in the same subnet, and between means, the communication between military entities of different subnets.

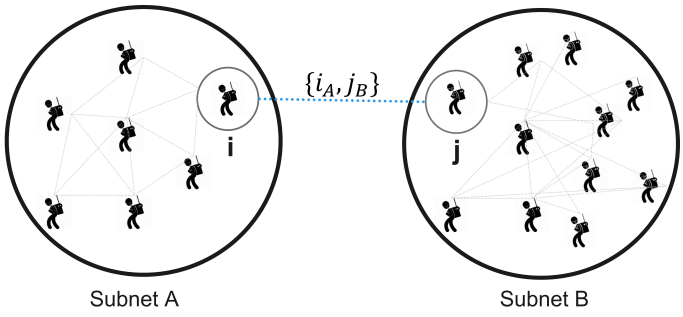


Figure 3: Link importance within and between the subnets.

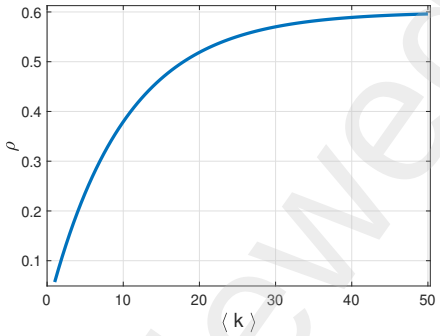


Figure 4: Relationship between successful transmission (ρ) and average degree of connectivity $\langle k \rangle$.

connectivity across military nodes. The second parameter, denoted as ρ, measures the successful transmission between them. To determine the relationship between these two parameters, we employ the nonperturbative heterogeneous mean field (npHMF) equation [29] as follows:

$$-\mu\rho + (1-\rho)(1-q) = 0 \tag{7}$$

where μ denotes the probability of transitioning from the transmission state to its non-transmission state (recovery). ρ denotes the successfully exchanged information between k-degree nodes, whilst q denotes the probability of those nodes that have not received any information from the other nodes. Its value can be further derived as follows:

$$q = (1-\beta\rho)^{\langle k \rangle} \tag{8}$$

By substituting the value of q from Equation (8) into Equation (7), we elegantly rewrite the entire expression in Equation (7) exclusively in terms of $\langle k \rangle$ as follows: as follows:

$$\langle k \rangle = \frac{\ln\left(1-\mu\frac{\rho}{(1-\rho)}\right)}{\ln(1-\beta\rho)} \tag{9}$$

It is evident from Equation (9) that the value of $\langle k \rangle$ and ρ are highly interrelated. To demonstrate their relationship using an example, Let us assume $0 \leq \rho \leq 1/(1+u)$. This implies that as the recovery rate μ increases, the successful transmission ρ decreases. For instance with $\mu = 0.5$, the maximum value of ρ would be around 0.667. Figure 4 shows the relationship between ρ and $\langle k \rangle$. This means that if a node has a higher degree of connectivity $\langle k \rangle$, it would have a higher chance to receive information through one of its links. By applying the definition of $\langle k \rangle$ derived in Equation (9) to Equations (4) $\sim$ (5), we can determine the link importance within the subnet as follows:

$$l_A \approx 2\beta^2\rho^A(1-\rho^A)^2\left(\frac{log(q^A)}{log(1-\beta\rho^A)}\right) \tag{10}$$

$$l_B \approx 2\beta^2\rho^B(1-\rho^B)^2\left(\frac{log(q^B)}{log(1-\beta\rho^B)}\right) \tag{11}$$

Similarly, we can also determine the link importance for information exchange between the subnets. For example, the link importance {*i*,*j*} in Figure 3 between node i belonging to subnet A, and node j belonging to subnet B can be derived as follows:

$$\begin{aligned} l_{ij} \approx\ & \beta^2\left[\rho^i(1-\rho^j)^2\left(\frac{\log(q^i)}{\log(1-\beta\rho^i)}\right)\right. \\ & \left.+\rho^j(1-\rho^i)^2\left(\frac{\log(q^j)}{\log(1-\beta\rho^j)}\right)\right] \end{aligned} \tag{12}$$

We can use Equation (12) to determine the link importance between the nodes of different subnets. This holds particular relevance in battlefield scenarios, for instance, such links could connect frontline fighters (infantry) with headquarters subnets. These connections are crucial, as they facilitate requests for support, be it aerial or naval assistance, and also enable the transmission of strategic commands from the headquarters to the frontline.

3.3. Determining the Critical Node

Within a MANET, a "critical node" refers to a specific node, If it encounters issues or fails, could significantly affect the overall performance or functionality of the entire network. A critical node can have at least one important link attached to it, as denoted by Equation (12). This equation is derived from the joint and conditional probabilities of information sharing in both directions[5]. In order to simplify the notation, we represent the joint probability as follows:

$$\phi_{ij} = P(\sigma_j = S, \sigma_i = I) \tag{13}$$

This implies that a higher value of ϕ_{ij} increases the likelihood of information being propagated from one node (e.g., military entity i) to another (e.g., military entity j). Given that any two nodes connected via a link have a high chance of propagating information when one of them already possesses the information, and if neither of them has the information but they are connected through one or more links, then all connected nodes have a susceptible probability of receiving the information. This can be expressed as:

$$\Theta^I_{ij} = P(\sigma_i = \sigma_j = I) \tag{14}$$

[5] As stated in Equation (3)

$$\Theta^S_{ij} = P(\sigma_i = \sigma_j = S) \tag{15}$$

Hence, the information received by a node from its connected links depends on the aforementioned three parameters, namely, ϕ, Θ^I, and Θ^S. The values of these parameters (ϕ, Θ^I, and Θ^S) undergo dynamic changes during battlefield scenarios, reflecting the consistent exchange of information among military entities. To estimate their anticipated values, we employ the information diffusion model of the susceptible-infected–susceptible (SIS) [30; 31; 29]. The probability of ϕ_{ij} at time $(t+1)$ can be evolved as follows:

$$\begin{aligned}\phi_{ij}(t+1) = \; & q_{ij}(t)\Big(1 - q_{ij}(t)\Big)\Theta^S_{ij} + \\ & \Big(q_{ij}(t)(1-\beta)\Big)(1-\mu)\phi_{ij}(t) + \\ & \mu\Big(1 - (1-\beta)q_{ji}(t)\Big)\phi_{ji}(t) + \mu(1-\mu)\Theta^I_{ij}(t)\end{aligned} \tag{16}$$

In this Equation, all possible states of information exchange and non-exchange between nodes i and j have been considered. The term $q_{ij}(t)$ denotes the probability that a specific node (i.e., i) is not exchanging any information with its neighboring nodes except for node j (i.e., unicast). This can be expressed as:

$$\begin{aligned}q_{ij}(t) &= \prod_{r=1,\ r\neq 1}^{N}\Big(1 - \beta A_{ri} P(\sigma_j = I | \sigma_i = S)\Big) \\ &= \prod_{r=1,\ r\neq 1}^{N}(1 - \beta A_{ri} h_{ri})\end{aligned} \tag{17}$$

where h_{ri} denotes the hostility function of node j towards node i. It measures the probability that node j is susceptible to receiving information from node i. The expansion of this function with respect to variables ϕ_{ij} and Θ^S_{ij} may be expressed as:

$$h_{ij} = \frac{\phi_{ij}}{\phi_{ij} + \Theta^S_{ij}} \tag{18}$$

Having gained an understanding of the function $q_{ij}(t)$ as described in Equations (17), we are now able to fully grasp all the components in Equation (16). The first term, i.e., $q_{ij}(t)\Big(1 - q_{ij}(t)\Big)\Theta^S_{ij}$; corresponds to the state where both nodes i and j are susceptible to exchanging information on the battlefield at time t. After some time (i.e., $(t+1)$), node j starts to receive or exchange information with its neighbors, while node i still remains susceptible to receiving it. the next term i.e., $q_{ij}(t)(1-\beta)(1-\mu)\phi_{ij}(t)$; represents the state where nodes i and j are susceptible to receiving information at time t, and at time $t+1$, both begin to receive the information. The third term, i.e., $\mu\Big(1 - (1-\beta)q_{ji}(t)\Big)\phi_{ji}(t)$; characterizes the state where node i begins to receive information from time (t) until time $(t+1)$. In contrast, node j undergoes a transition from not receiving information at time (t) to receiving it again at time $(t+1)$. The final term, i.e., $\mu(1-\mu)\Theta^I_{ij}(t)$; represents the state where both nodes i and j are exchanging information at time (t). However, at time $(t+1)$, node i undergoes a transition to not exchanging information, while node j continues to remain in a communication state.

By using the same procedure, we can also determine the anticipated values of $\Theta^I_{ij}(t+1)$, and $\Theta^S_{ij}(t+1)$ as follows:

$$\begin{aligned}\Theta^I_{ij}(t+1) = \; & \Big(1 - q_{ij}(t)\Big)^2\Theta^S_{ij}(t) + \\ & \Big(1 - (1-\beta)q_{ij}(t)\Big)(1-\mu)\phi_{ij}(t) + \\ & \Big(1 - (1-\beta)q_{ji}(t)\Big)(1-\mu)\phi_{ji}(t) + \\ & \mu(1-\mu)^2\Theta^I_{ij}(t)\end{aligned} \tag{19}$$

$$\Theta^S_{ij}(t+1) = 1 - \phi_{ij}(t) - \phi_{ji}(t) - \Theta^I_{ij}(t) \tag{20}$$

Equations (16), (19) and (20) define our system of 3L equations for determining the state of critical node. It should be noted that Equation (20) has been normalized, which results in the simplification of the $\Theta^S_{ij}(t+1)$ probability. To simplify Equations (16) and (19) further, we employ a linearization process. For This, we assume that the probability of information transfer between nodes denoted as ϕ_{ij}, ϕ_{ji}, and Θ^I_{ij}, $\ll 1$. Let $O(\epsilon)$ represent the smallest possible probability of information exchange between nodes through a link, thus, ϕ_{ij}, ϕ_{ji}, $\Theta^I_{ij} \sim O(\epsilon)$; and consequently $\Theta^S_{ij} \sim 1$ - $O(\epsilon)$. This linearized assumption would finally reduce Equations (16) and (19) as follows:

$$\begin{aligned}\phi_{ij} = \; & \beta\sum_{r=1}^{N}\Big(A_{rj} - 1 - (1-\mu)\delta_{ri}\Big)\phi_{jr} + (1-\beta) \\ & (1-\mu)\phi_{ij} + \mu(1-\mu)\Theta^I_{ij}\end{aligned} \tag{21}$$

$$\Theta^I_{ij} = \beta(1-\mu)\phi_{ij} + \beta(1-\mu)\phi_{ji} + (1-\mu)^2\Theta^I_{ij} \tag{22}$$

To solve 3L equations, they are first configured with an initial condition as $\Theta^I_{ij}(0) = \rho^2$, where ρ is the proportion of information spread, defined as $\rho_0(0 < \rho_0 \leq 1)$; $\phi_{ij}(0) = \phi_{ji}(0) = \rho_0(1-\rho_0)$. The iterative process continues until a fixed point is reached in the solution. Throughout the solution, the assumption is made that all nodes are susceptible to receiving information from any broadcasted node, implying $\Theta^S_{ij} = 1$.

As long as the probability of information exchange for a specific link stays below a critical threshold, the system preserves its equilibrium state. The threshold is determined by identifying the nontrivial solution of the system. When the transmission rate of a link surpasses that threshold, that specific node is recognized as the critical node responsible for conveying crucial pieces of information within the battlefield. The computation of the critical value is discussed in the following section.

3.4. Critical Threshold

The critical threshold serves as a value utilized to assess whether the information spread on a link within a Mobile Adhoc Network (MANET) is in an equilibrium state or an outlier state. To derive this threshold, we begin by examining the steady-state condition of the information spread probability. This does not imply a complete absence of information

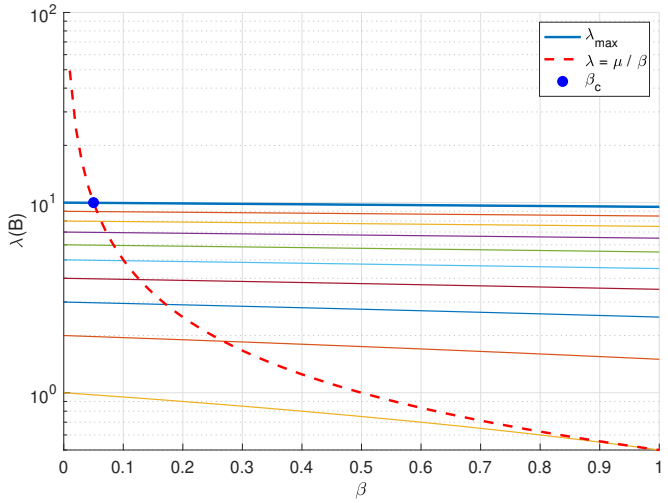


Figure 5: Critical link's Threshold in the MANET.

exchange between any two nodes; rather, there is still ongoing information exchange, but it occurs within a steady-state condition. It can be given by:

$$\Theta^I_{ij} = \frac{\beta(1-\mu)}{\mu(2-\mu)}(\phi_{ij} + \phi_{ji}) \tag{23}$$

This equation shows the dynamics of information transmission by considering both the success probability β and the recovery rate μ (from transmission to non-transmission state), while the joint probabilities account for the states of the nodes involved in the information exchange $(\phi_{ij} + \phi_{ji})$. Since, we aim to derive a threshold that can effectively distinguish the link state when there is communication happening. In order to do this, we assume that the communication between two nodes is at a minimum, denoted as follows:

$$\varepsilon_i = \phi_{ji} + \Theta^I_{ij} \ll 1 \tag{24}$$

This equation is independent of node j because the comprehensive likelihood of node i being in the Infectious state (i.e., exchanging information with its neighbors) is established by taking into account the various potential states of node j relative to node i. In other words:

$$P(\sigma_i = I) = P(\sigma_i = I, \sigma_j = I) + P(\sigma_i = I, \sigma_j = S) \tag{25}$$

In order to derive the link's threshold for information exchange between the critical nodes, we employ the linearized definitions of ϕ_{ij} and Θ^I_{ij} introduced in Equations (21) and (22) in the previous section. Substituting these values into Equation (24), we obtain the following result after simplification:

$$\frac{\mu}{\beta}\varepsilon_i = \sum_j B_{ij}\varepsilon_j \tag{26}$$

where B_{ij} is a matrix that is derived from the following components:

$$B_{ij} = (1-\Gamma)A_{ij} - \Gamma\delta_{ij}k_i \tag{27}$$

The B_{ij} matrix is the function of the constant Γ, which is independent of the link condition, and is defined as follows:

$$\Gamma = \frac{\beta(1-\mu)}{2\beta(1-\mu) + \mu(2-\mu)} \tag{28}$$

In summary, this constant provides a measure of the information spreading potential to persist or die out (i.e., its higher values indicates a greater likelihood of ongoing transmission within the population and vice versa). Moreover, A_{ij} in Equation (27) is the adjacent matrix of the connected links, k_i is the degrees of the node, and the δ_{ij} Kronecker delta function which is defined as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{29}$$

Figure 5 illustrates the μ/β plot, which decreases as β increases. To identify the critical node(s) within a MANET, it is necessary to determine the first eigenvalue of B_{ij} in Equation (26) that intersects with μ/β curve. The point of intersection, where they meet, represents the smallest value of β (associated with the largest eigenvalue of B). This critical value of the transmission probability, denoted as β_c, serves as the threshold for the critical node's transmission and is expressed as:

$$\beta_c = \frac{\mu}{\Lambda_{max}(B)} \tag{30}$$

where $\Lambda_{max}(B)$ denotes the maximum eigenvalue of matrix B. If a transmission probability exceeds this threshold, the nodes engaged in that link are considered critical nodes.

4. Framework Description

Herein, we first describe the problem formulation for detecting and jamming the critical nodes, and then propose our framework called RAID for achieving the desired objectives.

4.1. Problem Formulation

In a battlefield scenario, our objective is to jam the critical node(s) to prevent the spreading of information between the critical military entities. If V represents the total possible nodes defined as $V = \{v_1, v_2, ..., v_N\}$ and J is the critical set of nodes (that need to be jammed), i.e., $j = \{v_1, v_2, \ldots, v_k\}$, then our learning target is to minimize the accumulated risk of information spreading from the critical node(s). This accumulated connectivity is formalized as follows:

$$\min_{\{v_1, v_2, ..., v_k\}} = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{N}\frac{V \cap \{v_i \in J\}}{(v_j \in V)} \tag{31}$$

Here, v_i represents a node belonging to the critical node(s) that need to be jammed. To achieve equation 31, we consider two important measures.

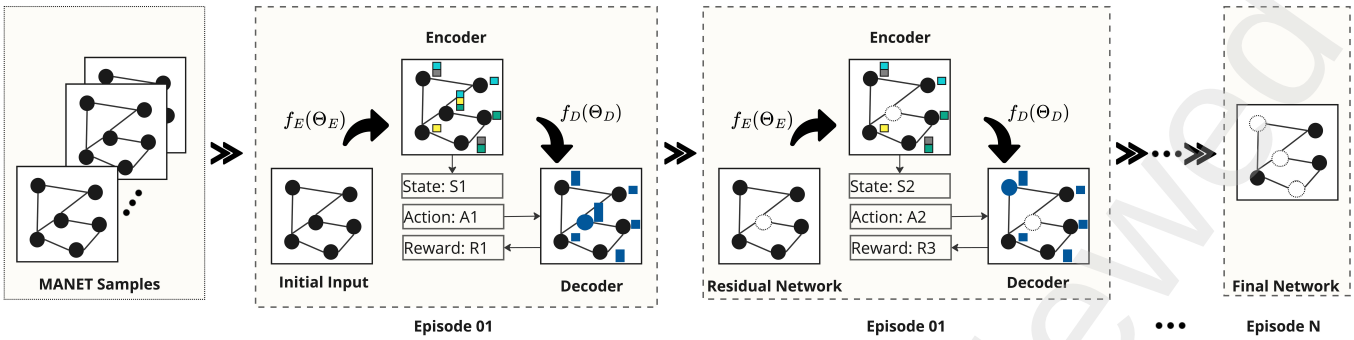


Figure 6: Offline training of MANET for detecting and jamming the communication of critical nodes.

1. **Critical Node Problem (CN):** If $\aleph$ is the whole network, and C_i represents the pairwise connected critical nodes within $\aleph$, then the sequence of all connected critical nodes is as follows:

$$Pairwise(C_i) = \sum_{C_i \in \aleph} \frac{\delta_i(\delta_i - 1)}{2} \tag{32}$$

where δ_i denotes the size of C_i.

2. **Network Dismantling (ND) Problem:** Given a cluster of C_i, the size of its nodes that must be dismantled to prevent the spread of information from the infected side network to the susceptible side network must be determined by finding the minimum cut set of C_i [32]. This minimum cut set consists of the smallest number of nodes that, when removed, will disconnect the infected nodes from the susceptible nodes, effectively halting the spread of information.

$$ND(\aleph) = min\{\delta_i, \forall C_i \in \aleph\} \tag{33}$$

4.2. Framework Architecture

RAID purely uses a data-driven approach and does not rely on domain-specific heuristics. It is based on a Markov decision process, where the MANET environment is traversed through a series of states, actions, and rewards. The state keeps updating the residual network[6], action identifies and removes the critical nodes responsible for conveying important information within the MANET whilst the reward tends to minimize the Equation (31). The model operates through a trial and error process, where its parameters are continuously updated as more episodes are processed, leading to increasing intelligence, as illustrated in Figure 6. It has two main important parts.

4.2.1. Encoder

The encoder utilizes graph embedding, which employs graph neural networks [33; 34; 35]. This graph representation learning approach is notably more resilient compared to conventional graph encoding methods like motif count [36], local degree distribution [37], and graphlet kernels [38] etc., which requires extensive feature engineering for node and graph representation. The encoder transforms the complex structure of a network (i.e., complex graph pattern) into a low-dimensional embedding vector. The embedding vector is computed for each node within the network by recursively aggregating features[7]. Through this iterative process, the embedding vector eventually integrates information about the node's interactions with its neighbors and its current position within the network.

The battlefield's MANET network resembles a graph, where the vertices ($\mathcal{V}$) represent mobile military entities, and the edges ($\mathcal{E}$) denote communication nodes between sets of nodes. Given this framework, the encoding process of encoder is shown in Figure 7 which begins by taking a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with input features $\{X_v \in \mathbb{R}^{1\times c}, \forall v \in \mathcal{V}\}$, depth K, and weight parameters $W_1 \in \mathbb{R}^{c\times p}$, $W_2 \in \mathbb{R}^{p\times(p/2)}$, and $W_3 \in \mathbb{R}^{p\times(p/2)}$. A virtual node s is created, which connects all nodes in the network. The virtual node s efficiently integrating global information and acts as a global context or graph state node. This enhances the model's ability to learn comprehensive and robust graph representations. Initially, the node representations $h_v^{(0)}$ are computed using $\sigma(X_v \cdot W_1)$, where σ introduces non-linearity and controls activation by zeroing negative values. Subsequently, these representations are normalized as $h_v^{(0)} \leftarrow \frac{h_v^{(0)}}{\|h_v^{(0)}\|_2}$ for all $v \in \mathcal{V} \cup \{s\}$. This normalization ensures consistent scaling which is crucial for enhancing convergence. For each layer l from 1 to K, and for each node $v \in \mathcal{V} \cup \{s\}$, the aggregated message from the neighbors $\mathcal{N}(v)$ is calculated as $h_{\mathcal{N}(v)}^{(l-1)} \leftarrow \sum_{j\in\mathcal{N}(v)} h_j^{(l-1)}$. The node representation is then updated using $\sigma([W_2 \cdot h_v^{(l-1)}, W_3 \cdot h_{\mathcal{N}(v)}^{(l-1)}])$ and normalized as $h_v^{(l)} \leftarrow \frac{h_v^{(l)}}{\|h_v^{(l)}\|_2}$. After K iterations, the final embedding vector for each node is $z_v \leftarrow h_v^{(K)}$. This process encodes the graph information into node embeddings by iteratively updating and combining node and neighborhood information.

4.2.2. Decoder

The decoder uses the state s and action a from the encoding process to compute a score function known as the Q function. The Q function predicts the maximum reward expected after taking action a in state s. This computation utilizes a two-layer multi-layer perceptron (MLP) with rectified linear unit (ReLU)

[6] i.e., the remaining links after removing the critical links in the previous episode. If it is the beginning, it will consider the links of entire network.

[7] e.g., the node's degree and the cost of its removal, etc.

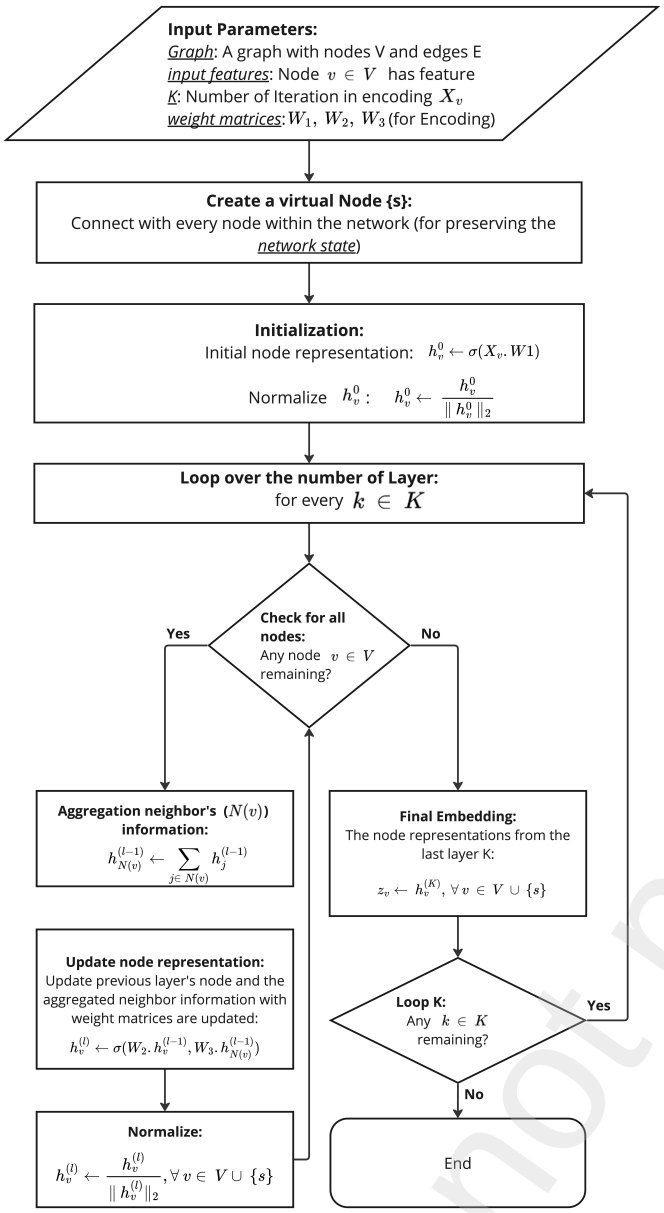


Figure 7: The encoder's process.

activation functions, structured as follows:

$$Q(s,a) = W_5^T.ReLU(z_v.W_4) \tag{34}$$

where W_4 and W_5 are the training phase weight parameters. The weight parameters in both the encoder and decoder are learnt by using n-step DQN [39]. $z_v \in \mathbb{R}^{1xp}$ is the output from encode which is equivalent to the corresponding state and action, i.e., $z_a^T.z_s$. The product operation between z_a^T and z_s is employed to capture intricate dependencies between them (i.e., state and action).

4.2.3. Model Training

The detailed training steps are illustrated in Algorithm 1. It begins by initializing crucial components: a replay buffer B to store past experiences, a Q-network with parameters Θ that learns to estimate Q-values for state-action pairs, and a target Q-network $\hat{\Theta}$ initialized with the same weights as Θ. Each episode starts by selecting a Mobile Ad-Hoc Network (MANET) with a random topology and initializing an empty state s_1. Within each episode, the algorithm iterates through time steps. Initially, the algorithm checks if it is in the initialization phase, where actions are taken for the first time. During this phase, actions are chosen randomly with a probability ϵ. However, once the algorithm has accumulated a history of prior actions, it switches to exploiting learned knowledge by selecting actions that maximize the Q-value for the current state s_t. After executing an action and observing the resulting reward, the algorithm updates the state to s_{t+1}. Periodically, experience replay stores transitions (sequences of states, actions, rewards, and next states) in the replay buffer B. These stored transitions are later sampled randomly to train the Q-network Θ. By using these samples, the algorithm updates Θ using gradient descent. This update process aims to minimize the error between the Q-values predicted by the network and the target Q-values, which are adjusted based on the observed rewards and the maximum Q-value predicted by the target Q-network $\hat{\Theta}$ for the subsequent state. Finally, after all episodes and time steps, the trained Q-network parameters Θ_Q are returned, representing the learned policy for optimally removing the critical node in dynamic environments.

5. Evaluation

In this section, we begin by detailing our adoption of Mobile Adhoc Networking within the Named Data Networking (NDN) architecture. Then, we present a comprehensive assessment across various MANET scenarios.

5.1. Adoptation of NDN in MANET

In battlefield scenarios where network topology is dynamic and nodes frequently move, data-centric model of Named Data Networking (NDN) simplifies data retrieval and management by allowing nodes to request data based on its content name [40; 41; 42; 43; 44]. This approach enhances efficiency by reducing redundant transmissions and leveraging data caching to improve access times [45]. The primary challenge with traditional Named Data Networking (NDN) is its design optimized for wired networks [46]. In contrast, battlefield scenarios require wireless infrastructure-less Mobile Ad-Hoc Networks (MANETs) that support mobility and dynamic topologies [24]. To address this challenge, this paper adopts NDN (utilizing existing NDN simulator referred to as ndnSIM [11; 12]) in MANETs using Network Simulator 3 (NS-3) to accommodate such battlefield environments.

The crucial layer where we apply modifications is the content-chunk layer [47]. This layer in ndnSim is responsible for segmenting large data objects into smaller pieces for easier transmission and retrieval. In particular, our modifications target the wireless communication part of the content-chunk layer. The default wireless module of this layer is shown in Figure 8(a). The `ForwardingStrategy` component determines

Algorithm 1: Training of Q-Network using Reinforcement Learning (RL)

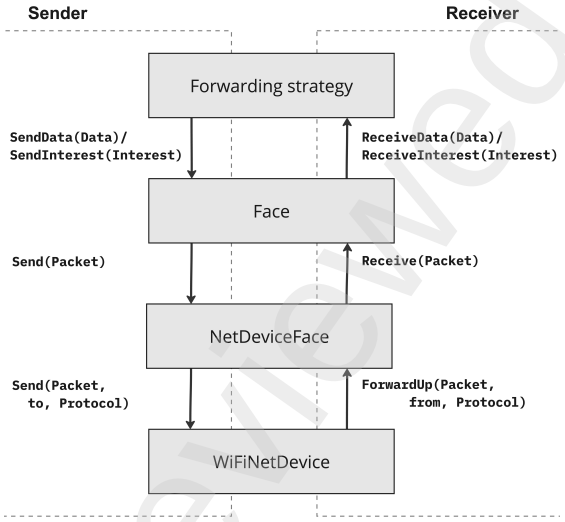
Input: Embedding vectors z_a, z_v; update frequency C; replay buffer size M; exploration rate ε; episode number N; time T; discount factor γ

Output: Trained Q network with parameters Θ_Q

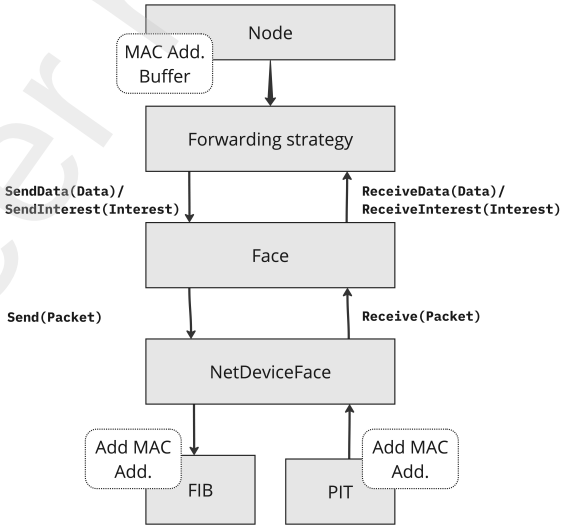
Initialization:
Initialize replay buffer B with size M;
Initialize Q network with parameters Θ;
Initialize target Q network with weights $\hat{\Theta} = \Theta$;
for *episode* = 1 *to* N **do**
  Select MANET with random topology;
  Initialize state $s_1 = ()$;
  for $t = 1$ *to* T **do**
    if *random action* $a_t == 1$ **then**
      Select a_t randomly with probability ϵ;
    else
      Select $a_t = \arg\max_a Q(s_t, a; \Theta)$;
    end
    Execute action a_t, observe reward r_t;
    Update state $s_{t+1} = s_t \cup \{a_t\}$;
    if $t \geq n$ **then**
      Store transition $(s_{t-n}, a_{t-n}, r_{t-n,t}, s_t)$ in B;
      Sample transition $(s_j, a_j, r_{j,j+n}, s_{j+n})$ from B;
      if s_{j+n} *is terminal state* **then**
        Set $y_j = r_{j,j+n}$;
      else
        Set $y_j = r_{j,j+n} + \gamma \max_{a'} Q(s_{j+n}, a'; \hat{\Theta})$;
      end
      Perform gradient descent update on Θ using the loss $(y_j - Q(s_j, a_j; \Theta))^2$;
    end
    if $t\%C == 0$ **then**
      Update $\hat{\Theta} = \Theta$;
    end
  end
end
return Q network parameters Θ_Q;

the best path for data and interest packets to travel through the network. The `Face`[8] class establishes connections between pairs of NDN nodes, whether they are producer/sender nodes, consumer/receiver nodes, or intermediate nodes. These connections are implemented using either TcpFace for TCP connections or UdpFace for UDP sessions. TcpFace ensures reliable, ordered data transmission, making it suitable for scenarios that demand strict reliability [48]. In contrast, UdpFace offers lightweight, best-effort delivery, prioritizing speed and efficiency. This setup enables direct exchange of Interest and Data packets between consumers and producers within the network.

[8]"Face" is preferred over "interface" because it includes not just the routing of packets across hardware network connections, but also the direct interaction of packets with application processes [47].



(a) Default ndnSim without MANET feature.



(b) ndnSim with MANET feature.

Figure 8: Changes applied to ndnSim in NS-3 for enabling the MANET feature, where **(a)** represents the default implementation and **(b)** represents the modified version with MANET.

The next class, `NetDeviceFace`, utilizes the `send()` method to transmit encapsulated packets either to a `WiFiNetDevice` for forwarding or to the `Face` for reception and decapsulation. It manages the conversion of packets into a suitable format for transmission across physical or virtual network devices. Lastly, the `WiFiNetDevice` serves as the network interface responsible for wireless communication, handling both the transmission and reception of packets over WiFi networks. In the default configuration of ndnSim, packets are broadcasted with the address `ff:ff:ff:ff:ff:ff` . This ensures that packets are sent to all devices on the network, mimicking a broadcast transmission.

To enable the MANET feature, we need to integrate two crucial features: (1) enable the content-chunk layer to adopt to the dynamic topology (due to frequent mobility in the battlefield scenario), (2) unicast the information to its target neighbor, rather than broadcasting to everyone. These enhancements

Table 1: List of hyperparameters used in the experiments.

Hyperparameter	Specified Value	Explanation
Maximum Episodes	1,000,000	The maximum number of training episodes
Replay Memory Size	500,000	Size of the experience replay buffer
Embedding Dimension	32	Dimension of the node embedding vector
Layer Iterations/Depth	3	Number of iterations for neighbor aggregation
Learning Rate	0.0001	Learning rate for the Adam optimizer
Time Update	1000	Frequency of updates for the target network
Q-learning steps	3	Number of steps in the multi-step Q-learning algorithm
Discount Factor	0.99	Discount factor (γ) used in the Q-learning update
Initial/Final Exploration	1/0.05	Initial/final values of ϵ in ϵ-greedy exploration
Exploration Steps	10,000	Steps to linearly reduce ϵ from its initial to final value
Mini-batch Size	32	Number of samples in each mini-batch during training

have been included in our modified ndnSim illustrated in Figure 8(b). We have enhanced the node component by introducing a MAC address buffer to cope with frequent node mobility and the resulting dynamic changes in network topology. This buffer allows nodes to efficiently store and manage MAC addresses of neighboring nodes, facilitating faster and more accurate routing decisions based on real-time network conditions. Additionally, we've integrated enhanced MAC address handling into the Pending Interest Table (PIT) and Forwarding Information Base (FIB). This update in the PIT and FIB ensure that Interest and Data packets are forwarded directly to their intended recipients through unicast transmissions, rather than being broadcasted to all nodes indiscriminately.

With these modifications, the existing ndnSim version 2.0 [11; 12] now supports MANET features, making it well-suited for simulating our battlefield scenarios for evaluation.

5.2. Simulation Results

All our experimental simulations were conducted on a Core-i9 HP desktop, equipped with 64 GB of RAM and an Nvidia GeForce RTX 3090 Ti graphics card with 24 GB of memory. Using the above-modified ndnSim framework in NS-3, we trained the model on random MANETs with small network nodes ranging from 50 to 100. To achieve this, we generated approximately 10,000 random networks and used them for training and around 100 instances for validation. We used TensorFlow 1.8 to implement our model and trained it using the Adam optimizer. The detailed list of our hyperparameters is shown in Table 1.

5.2.1. Assessment of Model's reliability

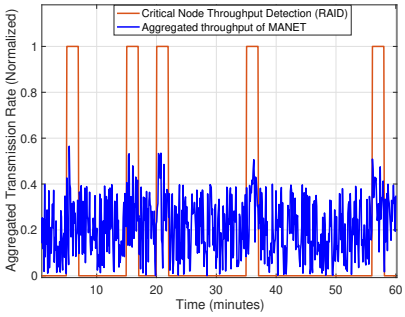
In a battlefield scenario, communication from critical nodes can begin at any moment, necessitating that our trained model effectively capture their transmissions. To evaluate this capability, we considered a topology of 100 nodes, with 90 being mobile MANET nodes and 10 being immobile, statically configured nodes. More Specifically, the MANET nodes, which are randomly distributed, where each node is transmitting to its neighbour node with a slightly lower throughput ($\leq$1 Mbps) with a random inter transmission frequency. The transmission between the MANET nodes is set randomly between 1 and 5 seconds[9]. In contrast, the static nodes (assumed to be the critical nodes) transmit at a slightly higher random rate (1-2 Mbps) and with a higher transmission frequency (1 to 2 seconds) during specific intervals: 5-7 minutes, 15-17 minutes, 20-22 minutes, 35-37 minutes, and 56-58 minutes within a one-hour timeframe. The objective was to determine if our model could efficiently detect the transmissions from these critical nodes, even during brief periods of aggressive transmission activity.

Figure 9(a) presents the aggregated transmission rate of the all nodes. It can be observed that during periods of critical nodes transmission, there is a slightly higher aggregated throughput for shorter durations (5-7 minutes, 15-17 minutes, 20-22 minutes, 35-37 minutes, and 56-58 minutes within a one-hour timeframe). The model accurately captures these transmissions during the specific intervals as depicted in red line. We also provide an assessment in Figure 9(b), which shows the identification of each individual critical node involved in the transmission along with its respective throughput. For a fair comparison, we have normalized the throughput values to a scale between 0 and 1. The time axis indicates the transmission time of every node whilst the node number is the index of each node involved in transmission. It is evident from Figure 9 that our model can effectively identify every critical node and capture its transmission instantly with accurate timing.

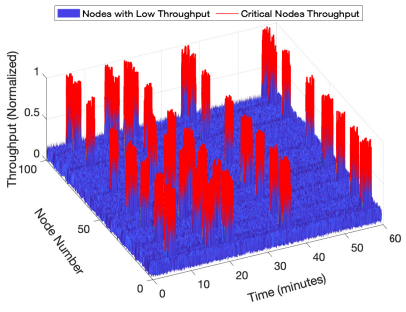
5.2.2. Assessment of Scalability Vs. Runtime

Scalability is a critical factor in battlefield MANET scenarios. In Figure 9(c), we analyze the runtime performance of our framework in a scalable MANET environment. We test network sizes ranging from 100 nodes to 10 million nodes. Furthermore, we evaluate five different cases with varying step ratios. The step ratio refers to the fraction of nodes removed at each adaptive step during network inferences. The curves exhibit a linear relationship relative to the number of edges. Notably, as we

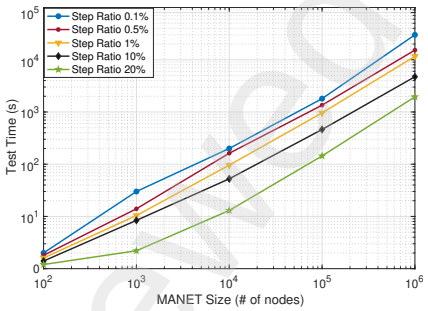
[9]For example, a node transmits 1 Mbps of data to its neighbor, then transmits another 0.5 Mbps of data after 3 seconds, and so forth. During this transmission, the throughput remains random (up to 1 Mbps), and the inter-transmission frequency is distributed randomly between 1 and 5 seconds.



(a) Detection of critical node throughput in a complex MANET scenario.



(b) Identification of critical nodes involved in information spreading.



(c) Scalability vs. Runtime.

Figure 9: **(a-b)** Reliability of our model for effectively detecting the critical nodes in a complex MANET scenario. **(c)** Runtime evaluation with different scale

increase the step ratio from 0.1% to 20%, the runtime for the network scale decreases (i.e., a larger step ratio—20%—results in a significantly reduced running time, approximately 159% lower than that of a 1% step ratio).

5.2.3. Assessment of Connected Node (CN) Problem

The purpose of this evaluation is to examine the problem of identifying a subset of critical nodes—whether weighted or unweighted—in a MANET, whose removal will help mitigate the spread of information among the remaining nodes in the network. For this assessment, we compare our approach with two key baselines: DeepELE [27], and FINDER [49]. DeepELE is a deep reinforcement learning-based framework that is intended to prevent the epidemic's spread. It automatically learns policies for identifying influential nodes (the nodes mainly responsible for spreading the infections) in a given graph. Similarly, the other baseline, FINDER, is a scalable deep reinforcement learning framework for identifying key players in complex networks using inductive graph representation learning to solve combinatorial problems. We adopted both baselines to our designated MANET and compared their performance with our framework.

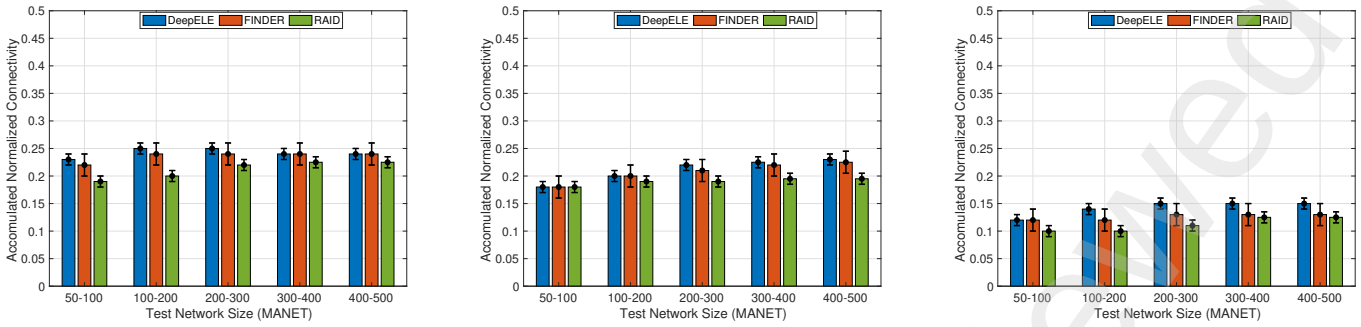
To assess the connected node problem, we generated a diverse range of MANETs with sizes ranging from 50 to 100, 100 to 200, 300 to 400, and 400 to 500 nodes, as illustrated in Figure 10. For each of the aforementioned scales, we generated 100 random instances using the Barabási–Albert (BA) model. In order to carefully examine, the connected node problem, we disabled the mobility feature of every node. All networks were handled as simplex (unidirectional) and any self-loops were eliminated. We examine three different cases for node weights: unweighted nodes, degree-weighted nodes, and randomly-weighted nodes.

Node Unweighted. In this case, no weights are assigned to any node. Figure 10(a) displays this case across all scales and compares the accumulated normalized connectivity metric (denoted by Equation – (31)) between the baseline and our framework. The accumulated normalized connectivity score evaluates the impact of removing critical nodes on the overall connectivity of a network. Each bar represents the average value calculated from 100 instances. It is evident that our framework results in a lower overall accumulated network connectivity score compared to others, indicating a more significant removal of critical nodes. This outcome is attributed to our framework's ability to effectively capture the connectivity between each node and its neighbors during the encoding process, regardless of the unweighted assignment. As an example, for the 200 to 300 scale, our framework achieves a mean score of 0.18, compared to 0.22 for FINDER and 0.23 for DeepELE. This indicates that our framework's removal of critical nodes is 20% more effective than FINDER and 24% more effective than DeepELE.

Node Degree Weighted. In this case weight is assigned to a node based on its degree, which is the number of connections (edges) it has to other nodes in the network. This weight does not reflect the node's connectivity or position in the network but rather introduces variability into the analysis presented in Figure 10(b). Our framework demonstrates a more intelligent approach to updating the degree weights of all nodes within the network compared to other baselines, achieving a lower accumulated normalized connectivity score for the considered scales.

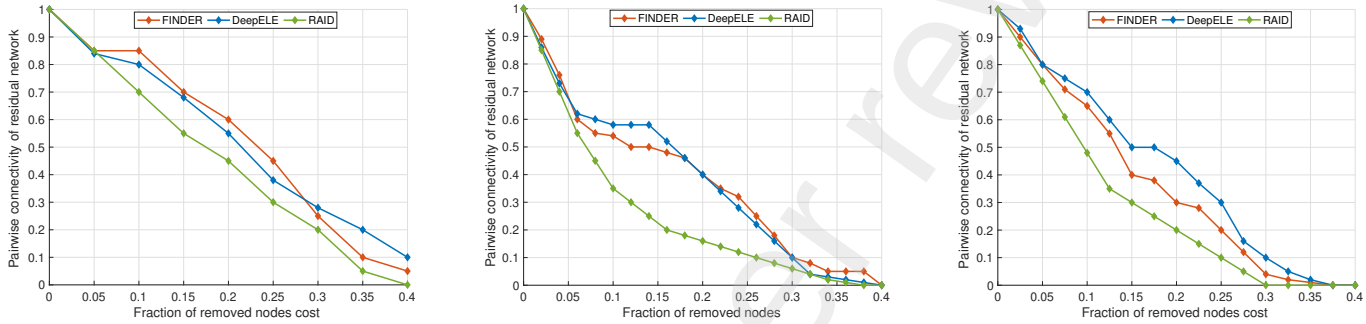
Node Random Weighted. Node random weight is calculated as $c(i) = 0.5 \times (d(i) + \bar{d} \cdot \epsilon)$, where $d(i)$ is the node's degree, $\bar{d}$ is the median degree of the network, and ϵ is a random variable from a normal distribution with mean 0 and variance 1. This formula combines the node's connectivity with a random factor scaled by the median degree, and the result is then halved[10] to determine the final weight. Using this weighted approach, the comparison between the baselines and our framework is illustrated in Figure 10(c). With the introduction of randomness, we not only examine the connectivity of individual nodes but also how random variations can influence the overall performance of the network. As a result, accumulated normalized connectivity metrics more accurately reflect the true influence of nodes by accounting for both their structural roles and inherent variability. This is evident from Figure 10(c), where our framework surpasses the two baseline approaches. This advantage stems from the careful weight updates shown in Figure 7, which enable our framework to compute accumulated normalized con-

[10]It moderates the overall influence of these factors



(a) Node Unweighted – Connected Node. (b) Node Degree Weighted – Connected Node. (c) Node Random Weighted – Connected Node.

Figure 10: Connected node problem with different cases **(a)** node unweighted, **(b)** node degree weighted, and **(c)** node random weighted



(a) Node Unweighted – Connected Node. (b) Node Degree Weighted – Connected Node. (c) Node Random Weighted – Connected Node.

Figure 11: Pairwise connectivity of residual network with different cases: **(a)** node unweighted, **(b)** node degree weighted, and **(c)** node random weighted

nectivity more efficiently. Moreover, the baseline approaches also demonstrate improvements compared to the two previously mentioned cases.

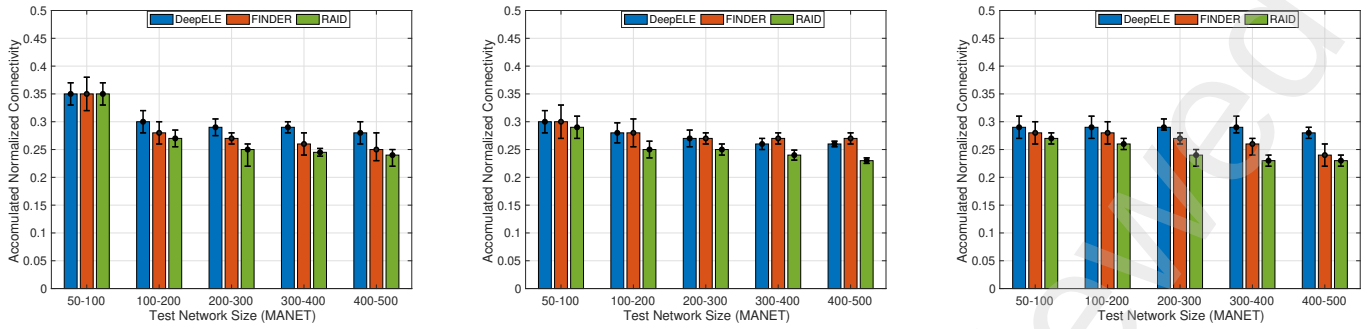
We further assessed the connected node problem by considering 100 instances of MANETs with 100 to 200 nodes for three cases: node unweighted, node degree weighted, and node random degree weighted. Figure 11 illustrates the pairwise connectivity of the residual network as a function of the fraction of removed nodes. We compare our framework with the two baseline approaches: FINDER and DeepELE. In Figures 11(a) to 11(c), all methods initially start with a pairwise connectivity of 1, which indicates full connectivity. As the fraction of removed nodes increases, the pairwise connectivity decreases for all approaches. It is noteworthy that compared to our framework, FINDER and DeepELE show a slower decline, meaning they maintain higher connectivity for a larger fraction of removed nodes. Conversely, our framework demonstrates a more rapid decline in pairwise connectivity, suggesting a higher effectiveness in identifying and removing critical nodes that significantly impact network connectivity.

Another aspect illustrated by Figures 11 is the overall trend showing how different node weighting strategies affect pairwise connectivity as nodes are removed. In all cases, connectivity decreases with an increasing fraction of removed nodes, but the rate of decline varies by method. For unweighted nodes on leftmost figure, the connectivity decline is less pronounced compared to other strategies. In contrast, with node degree weighting in the middle figure, the decline is steeper, indicating a more effective removal of critical nodes. Finally, in the rightmost figure, the decline in connectivity with random weights is the most significant. This indicates that this approach is more effective at targeting and disrupting critical nodes within the network.

5.2.4. Assessment of Network Dismantling (ND) Problem

In contrast to the connected node problem, which investigates the impact of node removal strategies on network connectivity, network dismantling aims to fragment the network into smaller components with minimal node removal. This approach assesses the network's resilience against critical node removals by measuring how connectivity degrades. For evaluating the network dismantling (ND) problem, we consider 100 instances for each scale in each node weighting case. Each bar represents the average result across these instances. The error bars indicate the deviation from the median value, with the upper cap showing the third quantile value and the lower cap showing the first quantile value.

Figure 12 compare the accumulated normalized connectivity resulting from the network dismantling of RAID to the baselines (DeepELE and FINDER) using different node weighting strategies. The comparison clearly demonstrates that RAID consistently achieves the lowest accumulated normalized connectivity across all scenarios, regardless of scale or weight variations. This lower accumulated normalized connectivity metric indicates that RAID is more effective at dismantling networks by significantly reducing their overall connectivity. In Figure 12(a), we evaluated network dismantling without assign-



(a) Node Unweighted – Network Dismantling (ND)).

(b) Node Degree Weighted – Network Dismantling (ND).

(c) Node Random Weighted – Network Dismantling (ND).

Figure 12: Node dismantling with different cases **(a)** node unweighted, **(b)** node degree weighted, and **(c)** node random weighted

ing weights to the nodes. RAID exhibits a notable reduction in connectivity across all scales, especially in smaller networks. When nodes are degree-weighted, as shown in Figure 12(b), the decline in connectivity improves further, leading to more effective removal of critical nodes. The most notable performance is observed in Figure 12(c) with random weights. In this case, RAID achieves the greatest reduction in connectivity, underscoring its effectiveness in efficiently targeting and removing critical nodes.

6. Discussion and Future Work

This paper is written from an adversarial perspective. It is focusing on disrupting (or jamming) the communication of critical nodes on the opposing side of the battlefield. We have designed a comprehensive mathematical model and a reinforcement learning framework to identify these critical nodes and subsequently disrupt their communication. With a thorough understanding of the adversarial side, we are now exploring a new direction to defend against such attacks. Our future work will focus on developing a counter-framework to protect and prevent against these disruptions.

As finding the critical node in a network is an NP-hard problem [35], we believe there is room for improving our existing model. We are exploring methods to enhance model accuracy and reduce computational complexity for large-scale network problems by investigating advanced unsupervised and semi-supervised convolutional learning techniques.

We also aim to implement our framework in realistic settings. This could be achieved by surreptitiously planting or sending a stealth device to the enemy side that is capable of inspecting ongoing wireless traffic at the MAC layer (e.g., by integrating a device like Air-Pcap [50; 51]) or at the physical layer (e.g., sniffing traffic in monitoring mode as in [52; 53] or using cognitive radio [54; 55]) to learn traffic patterns. Once the traffic is understood, our framework can identify the critical nodes. Various disruptive techniques (e.g., network flooding, poisoning the IP/MAC table, etc. [56]) can then be used to jam those target nodes. This will also be part of our future work.

7. Conclusion

In this paper, we present a framework known as RAID for removing critical nodes in battlefield scenarios to disrupt their communication. We have derived a comprehensive mathematical model for identifying the important links associated with critical nodes and formulated the problem of finding all critical nodes responsible for propagating information in the battlefield. We designed our framework referred to as RAID to address it. RAID employs a reinforcement learning method that incorporates an encoder and decoder. The encoder uses a graph embedding technique to transform the complex structure of a network into an embedding vector, aggregating features from the node neighbors, while the decoder assigns a score (maximum reward) to each embedding state and action. The trained model has been tested through comprehensive evaluations using our adopted Named Data Networking (NDN) MANET topologies. RAID has been assessed across various scales and weighting methods for both connected node and network dismantling problems. It outperformed existing RL-based baselines, achieving a 24% performance gain for smaller scale topologies (50-100 nodes) and 8% for larger scale topologies (400-500 nodes) in the connected node problem; and a 7% gain for smaller scale topologies and 15% for larger scale topologies in network dismantling problems.

Acknowledgement

This research was jointly supported by the research fund of Hanyang University under grant number HY-202400000001269 and by the Ministry of Trade, Industry, and Energy (MOTIE) under grant number RS-2023-00236325.

References

[1] Mohamed Younis and Sebnem Zorlu Ozer. Wireless ad hoc networks: technologies and challenges, 2006.

[2] Anders Fongen, Morten Gjellerud, and Eli Winjum. A military mobility model for manet research. *Parallel and Distributed Computing and Networks (PDCN 2009), February*, 16:18, 2009.

[3] Annamaria Paljanos, Simona Miclaus, and Calin Munteanu. Occupational exposure of personnel operating military radio equipment: measurements and simulation. *Electromagnetic Biology and Medicine*, 34(3):221–227, 2015.

[4] Jeman Park, Aziz Mohaisen, Charles A Kamhoua, Michael J Weisman, Nandi O Leslie, and Laurent Njilla. Cyber deception in the internet of battlefield things: Techniques, instances, and assessments. In *Information Security Applications: 20th International Conference, WISA 2019, Jeju Island, South Korea, August 21–24, 2019, Revised Selected Papers 20*, pages 299–312. Springer, 2020.

[5] Lin Zhu, Suryadipta Majumdar, and Chinwe Ekenna. An invisible warfare with the internet of battlefield things: A literature review. *Human behavior and emerging technologies*, 3(2):255–260, 2021.

[6] Christian Czosseck and Kenneth Geers. *The virtual battlefield: perspectives on cyber warfare*, volume 3. Ios Press, 2009.

[7] Steven Hildreth. Cyberwarfare. Congressional Research Service, Library of Congress, 2001.

[8] John V Blane. *Cyberwarfare: Terror at a click*. Nova Publishers, 2001.

[9] R. Sanchez, J. Evans, and G. Minden. Networking on the battlefield: challenges in highly dynamic multi-hop wireless networks. In *MILCOM 1999. IEEE Military Communications. Conference Proceedings (Cat. No.99CH36341)*, volume 2, pages 751–755 vol.2, 1999.

[10] Sahil Manchanda, Akash Mittal, Anuj Dhawan, Sourav Medya, Sayan Ranu, and Ambuj Singh. Learning heuristics over large graphs via deep reinforcement learning. *arXiv preprint arXiv:1903.03332*, 2019.

[11] Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. On the evolution of ndnsim: An open-source simulator for ndn experimentation. *ACM SIGCOMM Computer Communication Review*, 47(3):19–33, 2017.

[12] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnsim 2: An updated ndn simulator for ns-3. *NDN, Technical Report NDN-0028, Revision 2*, 2016.

[13] Michael R Frater and Michael J Ryan. *Communications Electronic Warfare and the Digitised Battlefield*. Land Warfare Studies Centre, 2001.

[14] WS Walton. *The Demos at Dawn: Marathon, 490 Bce*. Author House, 2008.

[15] Gilbert S Vernam. Cipher printing telegraph systems: For secret wire and radio telegraphic communications. *Journal of the AIEE*, 45(2):109–115, 1926.

[16] Brian N Hall. The'life-blood'of command? the british army, communications and the telephone, 1877-1914. *War & Society*, 27(2):43–65, 2008.

[17] TS Woolsey. Wireless telegraphy in war. *Yale Law Journal*, 14:247, 1904.

[18] Stephen Budiansky. *Battle of wits: the complete story of codebreaking in World War II*. Simon and Schuster, 2000.

[19] Barry M Wallack and George H Gero. Worldwide military command and control system (wwmccs) h-6000 tuning guide. volume i. wwmccs system tuning process. 1978.

[20] C Mark Melliar-Smith, Michael G Borrus, Douglas E Haggan, Tyler Lowrey, A San Giovanni Vincentelli, and William W Troutman. The transistor: An invention becomes a big business. *Proceedings of the IEEE*, 86(1):86–110, 1998.

[21] Christopher H Sterling. *Military communications: From ancient times to the 21st century*. Bloomsbury Publishing USA, 2007.

[22] Michael Russell Rip and David P Lusch. The precision revolution: The navstar global positioning system in the second gulf war. *Intelligence and National Security*, 9(2):167–241, 1994.

[23] Jack L Burbank, Philip F Chimento, Brian K Haberman, and William T Kasch. Key challenges of military tactical networking and the elusive promise of manet technology. *IEEE Communications Magazine*, 44(11):39–45, 2006.

[24] Cherukuri Rajabhushanam and Ayyaswamy Kathirvel. Survey of wireless manet application in battlefield operations. *International Journal of Advanced Computer Science and Applications*, 2(1), 2011.

[25] Biao Zhou, Kaixin Xu, and Mario Gerla. Group and swarm mobility models for ad hoc network scenarios using virtual tracks. In *IEEE MILCOM 2004. Military Communications Conference, 2004.*, volume 1, pages 289–294. IEEE, 2004.

[26] Juan G Restrepo, Edward Ott, and Brian R Hunt. Characterizing the dynamical importance of network nodes and links. *Physical review letters*, 97(9):094102, 2006.

[27] Peiyu Chen and Wenhui Fan. Identifying critical nodes via link equations and deep reinforcement learning. *Neurocomputing*, 562:126871, 2023.

[28] Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

[29] Sergio Gómez, Jesús Gómez-Gardenes, Yamir Moreno, and Alex Arenas. Nonperturbative heterogeneous mean-field approach to epidemic spreading in complex networks. *Physical Review E*, 84(3):036105, 2011.

[30] Yunpeng Xiao, Li Zhang, Qian Li, and Ling Liu. Mm-sis: Model for multiple information spreading in multiplex network. *Physica A: Statistical Mechanics and its Applications*, 513:135–146, 2019.

[31] Mei Li, Xiang Wang, Kai Gao, and Shanshan Zhang. A survey on information diffusion in online social networks: Models and methods. *Information*, 8(4):118, 2017.

[32] Alfredo Braunstein, Luca Dall'Asta, Guilhem Semerjian, and Lenka Zdeborová. Network dismantling. *Proceedings of the National Academy of Sciences*, 113(44):12368–12373, 2016.

[33] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[34] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11313–11320, 2019.

[35] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

[36] Chenhao Ma, Reynold Cheng, Laks VS Lakshmanan, Tobias Grubenmann, Yixiang Fang, and Xiaodong Li. Linc: a motif counting algorithm for uncertain graphs. *Proceedings of the VLDB Endowment*, 13(2):155–168, 2019.

[37] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on computing*, 21(1):193–201, 1992.

[38] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR, 2009.

[39] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[40] Alex Afanasyev, Jeff Burke, Tamer Refaei, Lan Wang, Beichuan Zhang, and Lixia Zhang. A brief introduction to named data networking. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6. IEEE, 2018.

[41] Tamer Refaei and Alex Afanasyev. Enabling a data-centric battlefield through information access gateways. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 634–639. IEEE, 2018.

[42] Ronald Doku, Danda B Rawat, Moses Garuba, and Laurent Njilla. Fusion of named data networking and blockchain for resilient internet-of-battlefield-things. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2020.

[43] Christopher Gibson, Pablo Bermell-Garcia, Kevin Chan, Bongjun Ko, Alex Afanasyev, and Lixia Zhang. Opportunities and challenges for named data networking to increase the agility of military coalitions. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6, 2017.

[44] Lorenzo Campioni, Mariann Hauge, Lars Landmark, Niranjan Suri, and Mauro Tortonesi. Considerations on the adoption of named data networking (ndn) in tactical environments. In *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–8. IEEE, 2019.

[45] Shahid Md Asif Iqbal et al. Adaptive forwarding strategies to reduce redundant interests and data in named data networks. *Journal of Network and Computer Applications*, 106:33–47, 2018.

[46] Toshihiko Kato, Ngo Quang Minh, Ryo Yamamoto, and Satoshi Ohzahata. How to implement ndn manet over ndnsim simulator. In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pages 451–456. IEEE, 2018.

[47] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12, 2009.

[48] Salman Muhammad, Touseef Javed Chaudhery, and Youngtae Noh. Study

on performance of aqm schemes over tcp variants in different network environments. *IET Communications*, 15(1):93–111, 2021.

[49] Changjun Fan, Li Zeng, Yizhou Sun, and Yang-Yu Liu. Finding key players in complex networks through deep reinforcement learning. *Nature machine intelligence*, 2(6):317–324, 2020.

[50] Ying Li, Yi Huang, Suranga Seneviratne, Kanchana Thilakarathna, Adriel Cheng, Guillaume Jourjon, Darren Webb, David B Smith, and Richard Yi Da Xu. From traffic classes to content: A hierarchical approach for encrypted traffic classification. *Computer Networks*, 212:109017, 2022.

[51] Junlin Yin and Syed Faraz Hasan. Passive localization for comparing physical activities in indoor environments. In *2022 International Conference on Information Networking (ICOIN)*, pages 352–355. IEEE, 2022.

[52] Muhammad Salman, Nguyen Dao, Uichin Lee, and Youngtae Noh. Csi: Despy: enabling effortless spy camera detection via passive sensing of user activities and bitrate variations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–27, 2022.

[53] Muhammad Salman, Lismer Andres Caceres-Najarro, Young-Duk Seo, and Youngtae Noh. Wisom: Wifi-enabled self-adaptive system for monitoring the occupancy in smart buildings. *Energy*, 294:130420, 2024.

[54] Miguel Camelo, Tom De Schepper, Paola Soto, Johann Marquez-Barja, Jeroen Famaey, and Steven Latré. Detection of traffic patterns in the radio spectrum for cognitive wireless network management. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[55] Fatima Salahdine and Naima Kaabouch. Security threats, detection, and countermeasures for physical layer in cognitive radio networks: A survey. *Physical Communication*, 39:101001, 2020.

[56] Hossein Pirayesh and Huacheng Zeng. Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey. *IEEE communications surveys & tutorials*, 24(2):767–809, 2022.