APPLICATION OF PROBABILISTIC INFERENCE TO RESILIENCY AND SECURITY ANALYSIS OF CYBER-PHYSICAL SYSTEMS

A Dissertation by

Ali Behfarnia

Master of Science, Iran University of Science and Technology, Iran, 2011

Bachelor of Science, University of Tabriz, Iran, 2008

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in the partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

May 2020

APPLICATION OF PROBABILISTIC INFERENCE TO RESILIENCY AND SECURITY ANALYSIS OF CYBER-PHYSICAL SYSTEMS

The following faculty members have examined the final copy of this dissertation for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Doctor of Philosophy, with a major in Electrical Engineering and Computer Science.

Ali Eslami, Committee Chair

Vinod Namboodiri, Committee Member

Sergio Salinas, Committee Member

Kaushik Sinha, Committee Member

Mehmet Bayram Yildirim, Committee Member

Accepted for the College of Engineering

Dennis Livesay, Dean

Accepted for the Graduate School

Coleen Pugh, Dean

DEDICATION

To my parents, Roya and Gholamreza, for their unconditional love, continuous support, and the inspiration of integrity, honesty, and hard work,

and

My sister and my brother who have been the sources of encouragement

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Prof. Ali Eslami, for his consistent help and support throughout my Ph.D. studies. I have learned plenty from his advice, and I feel that being a doctoral student under his supervision has been a special honor.

I also thank my committee members Prof. Sinha for being an endless resource of machine learning knowledge and helping me put observations in perspective; Prof. Salinas for encouraging me with unique brand-new ideas; Prof. Namboodiri and Prof. Yildirim for providing their time and suggestions.

I wish to extend my gratitude to my wonderful friends who have become a second family during my Ph.D. studies. My thanks goes to Reza Anthony Shayesteh, Khashayar Teimoori, Amirkhosro Kazemi, Meysam Ghanavati, Nam Nguyen, Mohammad Moulod, Omid Keivani, Farshad Houtaham, Yashar Zargari, and Seyed Ali Cheraghi. I am especially grateful for Sirvan Rahmati's frequent help and support during these years.

I would like to appreciate National Science Foundation and State of Kansas through the Kansas Board of Regents for their financial support in part under Grant OIA-1656006.

Finally, I would like to thank my parents who have always inspired me to pursue my goals and supported me during the ups and downs of my life. I would also like to thank both of my siblings for their endless love and support throughout these years.

ABSTRACT

This dissertation studies two important topics regarding the resiliency and the security of cyber-physical systems (CPSs). In the first work, a self-healing graphical representation is proposed to study the contagion of failures in self-healing interdependent networks. To this end, a graphical model representation of an interdependent cyber-physical system is proposed, in which nodes denote various cyber or physical functionalities, and edges capture the interactions between nodes. Then, a message-passing (belief propagation) algorithm is applied to this representation in order to analyze network reactions to initial disruptions. The framework is then extended to cases where the propagation of failures in the physical network is faster than the healing responses of the cyber network. Such scenarios are of interest in many real-life applications, such as the smart grid. As a result, it is proven that as the number of message-passing iterations increases, the network reaches a steady-state condition that would be either a complete healing or a complete collapse. The findings from this analysis help network designers have a better understanding of the resiliency of CPSs.

In the second work, security measurement and the malicious node detection of autonomous vehicles in intelligent transportation systems are studied. First, a simple security model based on Bayesian defense graphs is proposed to quantitatively assess the likelihood of threats against autonomous vehicles (AVs) in the presence of available countermeasures. Then, a game-theoretic model is represented using a local voting-based game to detect misbehaving neighboring vehicles in places where centrally managed stations are absent. In order to capture the inherent uncertainty of vehicles in ephemeral vehicular networks, a Bayesian game is used in which malicious nodes can potentially impact the result of the game. Then, equilibria of this game are obtained to study the strategies of malicious and benign nodes in networks. Using the analysis, the game parameters can be designed to achieve the maximum performance of misbehavior detection in vehicular networks.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Recent years have seen the growth of coupling between different systems to improve the efficiency of networks. A well-known example of an interdependent network is a cyber-physical system (CPS), where computational elements control physical entities. The future smart grid, intelligent transportation systems (ITSs), distributed robotics, and medical monitoring systems are all examples of CPSs. This work presents two challenges in CPSs. First, the problem of cascading failures in a self-healing CPS is addressed by employing the concept of belief propagation and low-density parity-check (LDPC) codes. To this aim, the reaction of CPSs is analyzed against an initial failure with respect to (w.r.t.) different network parameters. Second, security measurements and the misbehavior detection in ITSs are the focus. In particular, simple frameworks using Bayesian networks and Bayesian games are introduced to measure the security of autonomous vehicles (AVs) and to detect misbehavaring neighboring vehicles in ITSs.

A failure caused by a natural disaster or a malicious behavior could propagate from one network to another due to interdependency between CPS networks, and might end in massive economic and social disruptions. In 2003, a large-scale blackout in the northeastern United States led to millions of people losing their electricity for a few days [1]. The main cause of catastrophic failure was reported as the lack of proper training, modeling, and operations to respond to an urgent situation [2]. On the other hand, the *self-healing* capability of CPS provides a valuable opportunity, whereby the overlaying cyber network can cure failures in the underlying physical network. For instance, the traffic control network that monitors taxi transportation could avoid congestion by calculating the fastest routes during a given time of the day [3]. However, the modeling and design advantages of self-healing CPSs have been mostly overlooked in the literature, which provides the motivation for a holistic study

on detecting failures, and then applying protective models to predict and prevent large-scale cascading failures in CPSs.

This dissertation introduces a self-healing model for a CPS using a factor graph in which factor nodes denote various node functionalities, and the edges capture the interactions between nodes. Then, a belief propagation (a message-passing) algorithm is applied to study the dynamics of failure propagation and healing in this representation. The goal here is to investigate the resiliency of a self-healing interdependent network by understanding the transitions and steady-state behavior of a CPS against an initial distribution. It will be observed that a network reaches to a complete healing or a complete collapse as the number of message-passing iterations increases. Moreover, self-healing CPSs are studied for cases that the response of cyber nodes to failures in the physical network is delayed. Findings have revealed the crucial importance of low processing-time delay for increasing the probability of healing in the network. This work is comprehensively explained in Chapter 2.

The second work focuses on autonomous vehicles. Preventing accidents, saving time, and reducing traffic are just some advantages of autonomous vehicles that motivate many major manufactures, such as General Motors, Ford, and Volvo, to work on them [4]. In spite of major benefits, the safety and security of AVs are still among the main concerns. AVs are able to perceive the environment and maneuver without human action. This requires significant sensing and data processing that could make AVs vulnerable against cyber attacks and cause serious issues. Hence, it is of paramount importance to use countermeasures, and then assess provided security to study the reliability of AVs. On the other hand, vehicle-to-vehicle (V2V) communication is a feature that helps AVs plan ahead and make better decisions for maneuvering. However, information sharing between vehicles can be turned into a disadvantage, if a malicious vehicle sends data. Hence, it is vital to detect hostile vehicles on the roads, especially in places where central stations are absent.

This work addresses two problems for improving the security of future vehicles. First, a Bayesian network (BN) is developed to measure the security of AVs w.r.t. their employed

countermeasures. To do so, a directed acyclic graph is created to capture the cause-and-effect relationships between the elements of anti-attack techniques. Then, using standard scoring systems, the proposed model is evaluated to assess the security of a vulnerable component in AVs w.r.t. available countermeasures. Second, an attempt is made to detect misbehaving vehicles using collaboration between vehicles in places where centrally managed stations are absent. Taking ephemeral vehicular networks into considerations, a local voting-based game is developed to identify malicious nodes in the network. In the proposed design, proper incentives in expected utilities with the aim of identifying malicious nodes are offered to encourage cooperation. Also, the inherent uncertainty of vehicles regarding their monitoring systems, type of neighbors, and cost and benefit of cooperation are highlighted. In this context, Bayesian games are applied in order to capture the interactions between benign and malicious vehicles in the presence of uncertainties. These solutions are thoroughly described in Chapters 3 and 4.

1.1 Message-Passing Analysis of Self-Healing Interdependent Networks

The study of interdependent networks was sparked by the seminal work of Buldyrev et al. [5], where a simple "one-to-one" interdependence model was introduced. In this model, the coupling between cyber nodes and physical nodes was represented by bidirectional links between cyber and physical nodes, as shown in Figure 1.1. Based on this model, the transition phase of networks between a normal state and a large-scale collapse is analyzed using percolation theory. Many researchers have continued extending the findings of Buldyrev et al [5] to more realistic scenarios (a brief overview is provided in section 2.1). Among the most important issues on the way to designing future cyber-physical systems are the following: (i) derive an analytically tractable model that captures the key features of real-life systems such as self-healing and contagion, and (ii) develop a framework that enables studying multiple layers of interconnected cyber and physical systems. In this work, a novel approach is

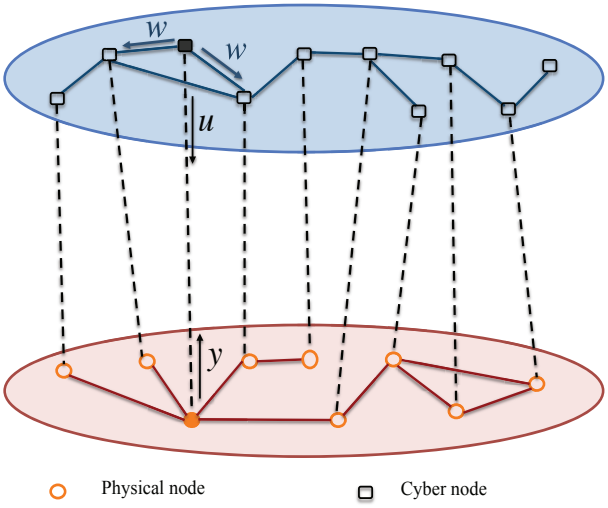


Figure 1.1: Illustration of "one-to-one" interdependent model.

taken to address these issues by applying ideas inspired by error-correction coding to model, analyze, and design CPSs.

First, a one-to-one interdependent model (Figure 1.1) introduced by Buldyrev et al. [5] is generalized using an extended graphical model representation, where nodes denote various cyber or physical functionalities, and edges capture the interactions between nodes. Then, belief propagation (a message-passing algorithm), which is used in low-density parity-check (LDPC) codes, is extended for the representation in order to study the dynamics of failure propagation and healing. By applying a density evolution analysis to this algorithm, network reaction to an initial disruption can be studied. It will be seen that as the number of message-passing iterations increases, the network reaches a steady-state condition, which would be either a complete healing or a complete collapse. To analyze the resiliency of a self-healing interdependent network, a sufficient condition on choosing the network parameters to completely heal the network after an initial disturbance is further obtained. Then, an optimization problem will be used to design cyber and physical networks for maximum resiliency. Next, an analytical framework where the propagation of failures in the physical network is faster than the healing responses of the cyber network will be developed. Such scenarios are of interest in many real-life applications such as the smart grid. Finally, exten-

sive numerical results are provided to verify the analysis and demonstrate the impact of the network parameters on the resiliency of the network.

1.2 Security Measurements for Intelligent Transportation System

Recent developments have made autonomous vehicles closer to being seen on our highways. However, their security is still a major concern among drivers as well as manufacturers. Hence, experts have continuously sought to identify gaps towards improving the security of AVs. Some researchers [6, 7, 8] have studied potential cyberattacks and their implications on automated and cooperative AVs. In particular, Petit and Shladover [6] categorized threats as high, medium, and low, based on some criteria used in the work of Stamatis [9], such as the feasibility of attack, the probability of attack success, etc. However, these threats against vulnerable components in AVs have not been quantitatively assessed. To do so, a simple security model based on Bayesian defense graphs is applied, in order to measure the likelihood of threats with respect to available countermeasures. As will be seen, different levels of security can be obtained, depending on what anti-attack techniques are employed.

Another important challenge is to detect misbehaving vehicles, especially where centrally managed stations are not present. Cooperation among vehicles has been a solution to detect malicious vehicles. However, vehicle uncertainty about their neighbors along with their monitoring system makes the problem difficult. In this regard, a game-theoretic model in a collaboration between vehicles, which captures the inherent uncertainty of vehicles about their monitoring systems, neighboring nodes, and the outcome of cooperation, is proposed. The goal here is to set up a approach to detect misbehavior nodes and limit malicious activities in vehicular networks.

1.2.1 Vulnerability Assessment of AVs Using Bayesian Defense Graph

The idea of attack graphs as a powerful tool is borrowed, which has already been used for computer network security. An attack graph is a graphical representation of all paths through a system that end in a state where an intruder successfully exploits the system.

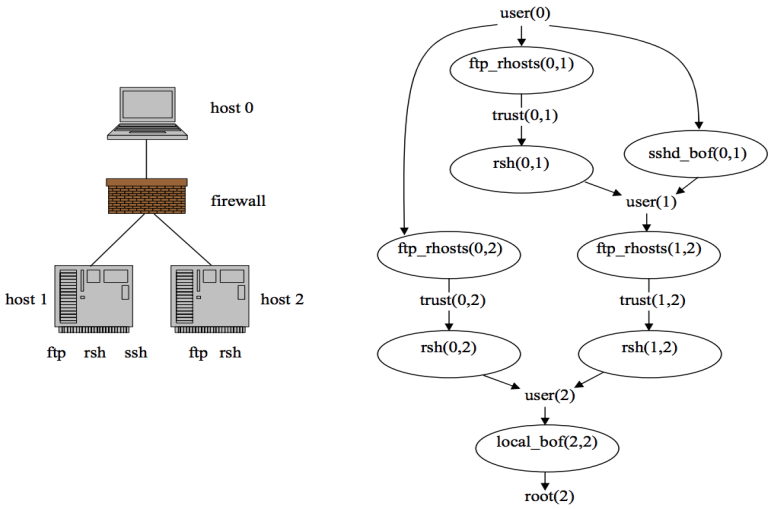


Figure 1.2: Example presentation of attack graph [10].

Figure 1.2 shows an example of an attack graph. A defense graph is formed similar to an attack graph, with the only difference being that the leaf nodes are countermeasures. Here, a Bayesian defense graph is applied to measure the security of vulnerable components in AVs. The results provide a belief about the degree of a vehicle's security, depending on available countermeasures. In a case study, the model and analysis for global positioning system (GPS) spoofing attacks are used to demonstrate the effectiveness of the proposed approach for a highly vulnerable component in an AV.

1.2.2 Cooperative Misbehavior Detection in Vehicular Networks

Game theory provides a powerful tool to analyze interactions between benign and malicious nodes in a network. In order to improve network performance, cooperation among vehicles can be utilized in places where road side units (RSUs) are not present. The local voting-based game is a cooperating method that can be employed to detect misbehavior vehicles in vehicular networks. Here, each node in a neighborhood sequentially broadcasts its votes w.r.t. an accused node (target node) with the aim of detecting faulty nodes. This voting game has been previously studied [11, 12, 13] to recognize malicious nodes and then temporarily evict them from the network. In these works, however, the uncertainty of vehicles, which is a major issue in AVs, has not been considered. Here, inspired by the voting

game, a game-theoretic model that captures the inherent uncertainty of vehicles about their monitoring systems, neighboring nodes, and the outcome of cooperation is proposed. In particular, one stage of a local voting-based game for identifying a target node is developed using a Bayesian game, while considering incomplete information of benign and malicious vehicles.

In order to analyze the proposed model and obtain equilibrium points, the expected utilities (payoffs) of malicious and benign nodes have been accurately designed based on critical features. One feature that greatly impacts the result of a cooperation is the selfishness of participants. Nodes might abstain from a collaboration while they benefit from its results, without making any contributions. These are known as free riders. By offering incentives in node payoffs, this problem can be addressed with the aim of increasing benign node participation. However, rewards should be adjusted according to the value of a contribution, which prevents nodes from abusing the benefit scheme by pointless participation. After designing payoffs, the model is analyzed using a pure-strategy Bayesian Nash equilibrium (BNE) and a mixed-strategy BNE to better understand the behavior of malicious and benign nodes in the network.

1.3 Organization

The rest of this dissertation is organized as follows. Chapter 2 studies the proposed self-healing interdependent network, where cyber nodes can heal failed physical nodes [14]. In particular, borrowing the concept of factor graphs, it will be shown how to apply a message-passing algorithm on a graphical representation to obtain steady-state behavior of a self-healing CPS [15]. Moreover, cases where propagation of failure in the physical network is faster than the healing response of the cyber network are studied in this chapter [16]. Chapter 3 is devoted to security measurements of autonomous vehicles using a Bayesian defense graph. A vehicle obtains a trustworthiness belief regarding its received signals, depending on what anti-attack techniques are being employed [17]. Chapter 4 considers the

uncertainties of vehicles in cooperation among vehicles, with the aim of detecting malicious nodes. In particular, a Bayesian game based on a local-voting game is developed to face those uncertainties in vehicles for misbehavior detection in ephemeral networks [18, 19]. Chapter 5 is devoted for future work wherein learning methods can be applied for the security of CPSs.

1.4 Publication

Below is a list of publications based on this dissertation.

1. A. Behfarnia and A. Eslami, "Misbehavior Detection in Ephemeral Networks: Local Voting Games in Presence of Uncertainty," in IEEE Access, vol. 7, pp. 184629-184642, 2019.

2. A. Behfarnia and A. Eslami, "Local Voting Games for Misbehavior Detection in VANETs in Presence of Uncertainty," in proc. IEEE 57 th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, Sep. 2019.

3. A. Behfarnia and A. Eslami, "Risk Assessment of Autonomous Vehicles Using Bayesian Defense Graphs," in proc. IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, Aug. 2018.

4. A. Behfarnia and A. Eslami, "Error Correction Coding Meets Cyber-Physical Systems: Message-Passing Analysis of Self-Healing Interdependent Networks," IEEE Transactions on Communications, vol. 65, no. 7, pp. 2753-2768, July 2017.

5. A. Behfarnia and A. Eslami," Dynamics and Steady-State Behavior of Self-Healing Cyber-Physical Networks in Light of Cyber-Node Delays," in proc. IEEE Globecom Workshops, Washington DC, Dec. 2016, pp. 1-6.

6. A. Behfarnia and A. Eslami, "Message Passing for Analysis and Resilient Design of Self-Healing Inter-Dependent Cyber-Physical Networks," in proc. 25th IEEE International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, Aug. 2016, pp. 1-6.

CHAPTER 2

MESSAGE-PASSING ANALYSIS OF SELF-HEALING INTERDEPENDENT NETWORKS

Coupling cyber and physical systems gives rise to numerous engineering challenges and opportunities. An important challenge is the contagion of failure from one system to another, which can lead to large-scale cascading failures. However, the *self-healing* ability emerges as a valuable opportunity where the overlaying cyber network can cure failures in the underlying physical network. To capture both self-healing and contagion, this work considers a graphical model representation of an interdependent cyber-physical system, in which nodes represent various cyber or physical functionalities, and edges capture the interactions between the nodes. A message-passing algorithm is proposed for this representation to study the dynamics of failure propagation and healing. By conducting a density evolution analysis for this algorithm, network reaction to initial disruptions is investigated. It is proved that as the number of message-passing iterations increases, the network reaches a steady-state condition that would be either a complete healing or a complete collapse. Then, a sufficient condition is derived to select the network parameters to guarantee the complete healing of the system. The result of the density evolution analysis is further employed to jointly optimize the design of cyber and physical networks for maximum resiliency. This analytical framework is then extended to the cases where propagation of failures in the physical network is faster than the healing responses of the cyber network. Such scenarios are of interest in many real-life applications such as smart grid. Finally, extensive numerical results are presented to verify the analysis and investigate the impact of the network parameters on the resiliency of the network.

The rest of this chapter is organized as follows. Section 2.1 provides a brief overview of the related work. Section 2.2 provides a brief introduction to the use of message passing and density evolution in graphical models, particularly factor graphs. Section 2.3 starts the

analysis by applying a message passing algorithm to a simple self healing one-to-one network inspired by Buldyrev's model. Then, section 2.4 describes the system model, notations, and message passing in the general CPS. Section 2.5 provides a density evolution analysis of the proposed message-passing algorithm. This section also includes a sufficient condition for the system to be healed, optimizing network parameters for maximum resiliency, and the effect of processing time delay in the network. Section 2.6 is devoted to extensive numerical results, and Section 2.7 concludes this chapter.

2.1 Related Works

Several papers extended the findings of Buldyrev et al. [5] by applying percolation theory while focusing on the size of the remaining giant component after a cascading failure. Authors in [20] studied the percolation of failures after an attack in a one-to-one interdependent network model in which mutually dependent nodes have the same number of neighbors. Parshani et al. [21] studied the case where only a fraction of the nodes in both networks depend on each other, that is, some nodes in each network are not connected to the other network. They proved that the reduction of coupling between networks leads to a change from a first-order percolation phase to a second-order percolation phase. Later, a systematic strategy based on betweenness was introduced [22] to select a minimum number of autonomous nodes that guarantees a smooth transition. This reduces the fragility of the network without losing functionality. Shao et al. [23] proposed an interdependent model, taking into account the realistic scenarios at which a node in a network X might be supported by more than one node in a network Y, and vice versa. In such cases, a node will continue to work as long as at least one of its supporting nodes is still working.

Gao et al. [24] have developed an analytical framework for studying the robustness of tree-like n interdependent networks. They found that for any $n \geq 2$ (for Erdös-Rényi, random regular, and scale-free graphs), cascading failure appears, and transition becomes a first-order compared to a second-order transition. A "regular allocation" algorithm was proposed in [25]

to allocate the same number of interlinks to each node. Authors proved that such allocation is optimal for a network with an unknown topology, and that employing bidirectional interlinks instead of unidirectional ones leads to better robustness. In a different line of work, several authors [26], [27] studied the influence of active small clusters appearing after an attack on the whole network performance. In particular, they obtained an upper bound for the fraction of operating active small clusters after a cascading failure. Shahrivar et al. [28] studied the resilience of random interdependent networks through algebraic connectivity. They obtained a threshold for r-robustness, which is the same as that required for the graph to have a minimum degree r.

A number of works [29, 30, 31, 32, 33, 34] have been devoted to self-healing single-layer networks. Several authors [31, 30] studied the concept of self-healing networks through distributed communication protocols that set up new links to recover the system connection. In particular, Quattrociocchi et al. [31] evaluated the performance of redundant links in small-world networks against link failures. Through small-world topologies, they found that some long-range connections could greatly increase the resiliency of the network. Liu et al. [32] investigated the effect of restoration time and resources to study the cascading overload failure in homogeneous (e.g., Erdös-Rényi) and heterogeneous (e.g., scale-free) networks. For an SIS-type epidemic process, Drakoloulos et al. [33] achieved a lower bound on the optimal expected extinction time. This bound was obtained as a function of curing budget, maximum degree of each node, and epidemic parameters. Besides, many real-time technologies in realistic networks that support the realization of self-healing methods have been developed. For example, Li et al. [34] developed an optimization problem for a protection strategy (e.g., switching transmission line) in a critical infrastructure (e.g., power grids).

A few works [35, 36] have recently studied self-healing multi-layer networks. Stippinger et. al [35] introduced a healing strategy for an interdependent network based on the formation of new links. By applying recovering links after failures, they found that the increase in resiliency of an interdependent network has power-law scaling with the probability

of healing. They also showed that it is possible to suppress the cascading failure by keeping the healing probability above a critical value. Majdandzic et al. [36] developed a phase diagram for multi-layer networks to find an optimal repairing strategy in damaged interacting systems.

Our work is concerned with the study of failure, the evolution of failure, and the recovery process in a CPS. The main differences between this work and the above literature are as follows: i) we apply, for the first time, a message-passing analysis to study the dynamics of failure propagation and healing in a cyber-physical system, ii) we obtain a closed-form equation for the evolution of failure in a self-healing interdependent network, iii) we develop a closed-form equation for the evolution of failure in the presence of time delay for recovering nodes in a self-healing interdependent network, and iv) we derive a sufficient condition for a self-healing cyber-physical system that prevents cascading failure in the network. In all the above, we exploit techniques from coding theory to analyze network resiliency.

2.2 Message Passing and Density Evolution in Graphical Models

This section provides a background on message passing in factor graphs, which is required to understand the analysis in this work. We start with a brief introduction to factor graphs and their message passing. We then explain message passing in the simple example of LDPC decoding, followed by a description of density evolution as it is done in codes on graphs. Finally, we present the similarities and differences between codes on graphs and CPSs in their factor graphs, message passing, and density evolution.

2.2.1 Factor Graphs

A factor graph could be defined as a bipartite graph that expresses the factorization of a function. To explain this, let $g(x_1, x_2, ..., x_n)$ be a function that could be factored into product of several local functions, each having a subset of all variables, $\{x_1, x_2, ..., x_n\}$. So,

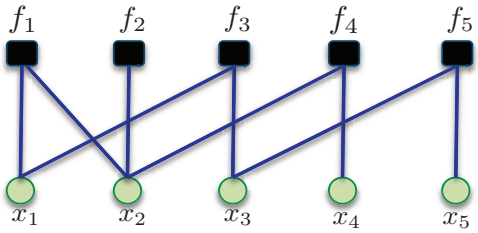


Figure 2.1: A factor graph for the product $f_1(x_1, x_2)f_2(x_2)f_3(x_1, x_3)f_4(x_2, x_4)f_5(x_3, x_5)$.

we have

$$g(x_1, x_2, ..., x_n) = \prod_{k \in K} f_k(X_k) \tag{2.1}$$

where K is a discrete index set, X_k is a subset of $\{x_1, x_2, ..., x_n\}$, and $f_k(X_k)$ is a function with the elements of X_k as arguments. A *factor node* denotes a local function f_k, a *variable node* denotes each variable x_k, and an edge exists between variable node x_k and factor node f_k if and only if x_k is an argument of f_k. Figure 2.2.1 shows an example of a function, $g(x_1, x_2, x_3, x_4, x_5)$, on a bipartite graph that can be obtained as the product of $f_1(x_1, x_2)f_2(x_2)f_3(x_1, x_3)f_4(x_2, x_4)f_5(x_3, x_5)$.

A popular message-passing algorithm on factor graphs is the sum-product message-passing algorithm, also known as belief propagation, which computes all marginals of the individual variables of the function. Computation of a marginal function begins with the leaves of a factor graph. Each leaf variable node and leaf factor node send "belief" messages to their parents. A variable node simply sends the product of all received messages as its belief, while a factor node with parent x calculates the product of all received messages from its children, and then operates on the result with a sum over all its variables except x, $\sum_{\sim\{x\}}$, to send its belief. It is worth noting that the role of parents and children nodes are temporary, and it would be variant for different marginalized parameters. To obtain a mathematical expression for the message-passing algorithm, let $n(y)$ denote a set of all neighbors of a node y, $\mu_{f\to x}(x)$ represents a message sent from a factor node, f, to a variable node, x, and $\mu_{x\to f}(x)$ shows the message sent from node x to f. Then, as illustrated in

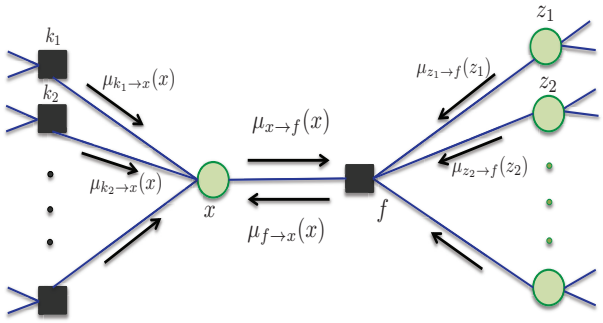


Figure 2.2: A part of factor graph, showing the update rules of sum-product message-passing algorithm.

Figure 2.2.1, the sum-product message-passing algorithm can be written as follows [37]:

$$\mu_{x\to f}(x) = \prod_{k\in n(x)\backslash\{f\}} \mu_{k\to x}(x) \tag{2.2}$$

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{z\in n(f)\backslash\{x\}} \mu_{z\to f}(z) \right). \tag{2.3}$$

For example, $g(x_4)$ can be obtained as follows:

$$g(x_4) = \mu_{f_4\to x_4}(x_4) \tag{2.4}$$

where,

$$\mu_{f_4\to x_4}(x_4) = \sum_{\{x_2\}} f_4(x_2, x_4)\, \mu_{x_2\to f_4}(x_4) \tag{2.5}$$

$$\mu_{x_2\to f_4}(x_4) = \mu_{f_2\to x_2}(x_2)\, \mu_{f_1\to x_2}(x_2). \tag{2.6}$$

2.2.2 Message Passing in Codes on Graphs

Message passing has been successfully employed in the decoding of codes on graphs. Here we explain the message-passing decoding of LDPC codes with a simple example. Figure 2.3 shows part of the *Tanner graph* of an LDPC code, where the circles and squares represent, respectively, *variable nodes* and *check nodes*. The variable nodes correspond to the symbols

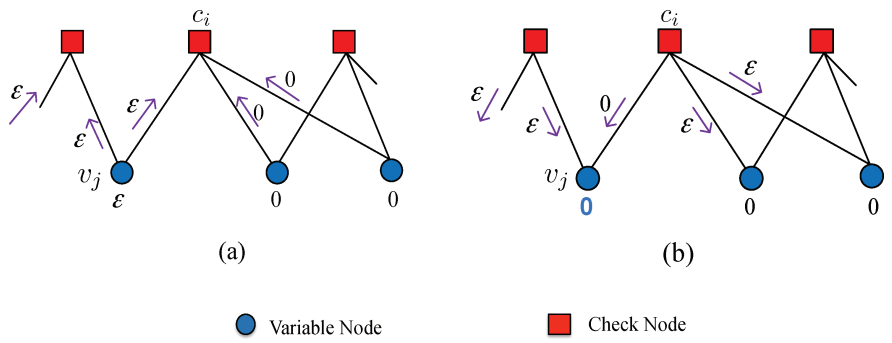


Figure 2.3: One iteration of message passing in LDPC codes over binary erasure channel.

received from the channel, i.e., channel outputs. In this particular example, we assumed a *binary erasure channel* (BEC) where the channel outputs are either received correctly or are unknown. In a BEC, there is no bit flip from 0 to 1, or vice versa. The functionality of a check node is to do a check-sum on the values of its variable nodes and ensure that they add up (in modulo 2) to zero.

The goal of the decoder is to determine the correct values of the unknown bits after multiple rounds of message passing between the variable and check nodes. As shown in Figure 2.3, there is one unknown bit (variable node) at the channel output. At the beginning of each iteration, every variable node v_j sends its value to all of its neighboring check nodes, as shown in Figure 2.3 (a). Every check node c_i then derives what it believes about the value of each of its neighboring variable nodes, and sends the information back to each of them as a message. To derive this value for v_j, c_i uses the messages received from all of its variable nodes, excluding v_j. If one or more of these messages are ϵ (erasure), then c_i cannot be of help to v_j at this round and therefore sends an ϵ to v_j. Otherwise, if all of these messages are either 0 or 1, then c_i takes their check-sum and sends the result to v_j as what it believes v_j should be. An example of messages sent from check nodes to variable nodes is shown in Figure 2.3(b). The last two steps are repeated until the values of all variable nodes are derived, or a certain number of iterations is reached.

2.2.3 Density Evolution in Codes on Graphs

For the message-passing decoding of codes on graphs on general memoryless channels, messages between variable nodes and check nodes are often defined as log-likelihood ratios (LLRs) of probabilities that a given bit is "1" or "0". Since LLRs are often continuous variables, the probability of a message for a specific value of LLR can be described by a probability density function (pdf). Tracking the evolution of this pdf in a message-passing decoder is called *density evolution* (DE) and can reveal the performance of the decoder. While DE is typically used for channels like binary additive white gaussian noise channel (BIAWGNC) with continuous LLRs, this term can also be employed to study the evolution of erasures in BEC channels where LLRs are discrete. In this case, DE keeps track of the density of erasure messages (ϵ) to analyze the performance of the decoding algorithm. For this, let p_i denote the probability of ϵ message from a variable node to a check node, and let q_i denote the probability of ϵ from a check node to a variable node, both in the i-th iteration of message passing in Figure 2.3. The probability of ϵ message on the $(i+1)$-th iteration of message passing from a variable node with degree d, say v_j, to a check node, say c_t, can be written as $p_{i+1} = p_i\, q_i^{d-1}$ under the independence assumption [38]. This equation is actually obtained using equation (2). The term q_i^{d-1} accounts for all received messages to v_j except c_t (to avoid positive feedback) which is going to receive a message $\prod_{k \in n(x)\setminus\{f\}} \mu_{k\to x}(x)$. It was proved in [38] that the decoding algorithm is successful if the inequality $p_{i+1} < p_i$ holds for every $i \geq 0$.

The recursive equation above is referred to as "density evolution equation", and was obtained for message passing in LDPC codes over a BEC. Similar equations can be derived for message-passing algorithm over other channels and for other types of codes on graphs (see [39, 40, 41]). In this work, we apply the density evolution analysis to the factor graph of CPSs, and derive the density evolution equation for them.

2.2.4 Codes on Graphs vs. Cyber-Physical Systems

There are a few similarities and differences between codes on graphs and cyber-physical systems in the structure of their factor graphs and properties of messages. In codes, factor nodes check/correct variable nodes; in CPSs, cyber nodes can *heal* physical nodes. Similar to the codes, there are two types of nodes in a CPS, and their interactions can be captured through messages exchanged between them. However, some differences between the two applications can be recognized as follows.

1. In codes on graphs, unknown (damaged) variable nodes cannot affect the functionality of the check nodes. In other words, *failure* cannot propagate from variable nodes to check nodes. On the other hand, in cyber-physical systems, failure of the physical components could cause a failure in the cyber network, and vice versa [42].

2. Factor nodes in a CPS factor graph may assume a more complicated functionality than in codes on graphs. Also, the delivery of the messages in a CPS may not be guaranteed. These result in a different message structure and, probably, a more complicated density evolution equation.

3. The Tanner graph of a code is a bipartite graph in which every edge connects a check node to a variable node. In other words, there are no edges connecting the variable nodes or check nodes. In a CPS, however, both of the cyber and physical systems are connected networks. Hence, a physical (cyber) node can directly affect the operation of other physical (cyber) nodes.

In this chapter, we show how message passing can be applied to CPSs despite these differences. The first two differences can be addressed by appropriately defining the messages, and physical and cyber node functions, while accounting for imperfect message passing. This will be presented in sections 2.4 and 2.5. The last difference can be addressed using a rather standard approach, i.e., by adding virtual check nodes to the CPS factor graph. This will be presented in details in the proof of Theorem 2 in section 2.5. To provide intuition into our approach, we start by considering a slightly modified version of Buldyrev et al.'s "one-

to-one" network [42]. We will then extend our analysis to more general cases of CPSs and include cyber-node time delays.

2.3 Message Passing over a Self-Healing One-to-One Model

Buldyrev et al. [42] introduced a simple "one-to-one" model that yields important insight into studying interdependent networks. In this model, it is assumed that two networks, say A and B, have the same number of nodes, N. The state (failed or alive) of a node in network A directly depends on the state of the

corresponding node in network B. Figure 2.4 shows such a one-to-one model of a cyber-physical network. There will be an initial attack on the physical network failing each physical node with a probability ϵ. Failures then propagate, not through the physical network but from the physical nodes to the cyber nodes and then through the cyber network. A cyber node with only failed cyber neighbors will fail, hence failing its underlying physical node. As time passes and failure propagates between the two networks after several *iterations*, a catastrophic cascade of failures may occur.

In the model of Buldyrev et al., if a physical node fails, then the corresponding cyber node will also be lost, and there is no healing capability for either physical or cyber nodes. We slightly modify this model to consider a healing ability for cyber nodes. We assume that a cyber node that is not isolated from the cyber network can heal its failed physical node. That is, a cyber node with at least one healthy cyber neighbor still has access to the cyber network's data and can heal its physical node.

We capture the propagation of failure and healing between the nodes as *defection* (D) and *healing* (H) *messages* exchanged between them, and apply message-passing analysis tools to study the evolution of the cascade in this interdependent network. We may look at this evolution within one (any) iteration, and see how the failure probability changes for physical nodes. If this probability increases at the end of the iteration, then a cascade will occur, and if it decreases at the end the iteration, then the network will heal completely.

Let us consider the first iteration after the initial attack. Each physical node is failed at the beginning with probability ϵ. Let y denote the probability of a D message from a physical node to its cyber neighbor. Thus,

$$y = \epsilon. \tag{2.7}$$

A cyber node with a failed physical node sends D messages to all its cyber neighbors reporting that it has lost its physical connection. Denote the probability of this event by w. Since each cyber node is connected to only one physical node,

$$w = y = \epsilon. \tag{2.8}$$

A cyber node with only failed cyber neighbors sends a D message to its physical node; otherwise, it sends an H message healing the physical node. If we denote the probability of the former (sending a D message to the physical node) by u, then

$$u = \rho(w) = \rho(\epsilon), \text{where} \quad \rho(z) = \sum_{i \geq 1} \rho_i z^i \tag{2.9}$$

is the degree distribution of the cyber nodes, and $\sum_{i \geq 1} \rho_i = 1$. Here, ρ_i is the fraction of cyber nodes with i cyber neighbors. Let us denote the probability of a physical node failure by x:

$$x = \Pr\left\{\text{Receiving a D message from the cyber node}\right\} \tag{2.10}$$

Also, assume that x_l represents the l-th iteration of message passing. Now, we are at the end of the first iteration and x_1 can be written as:

$$x_1 = u = \rho(\epsilon). \tag{2.11}$$

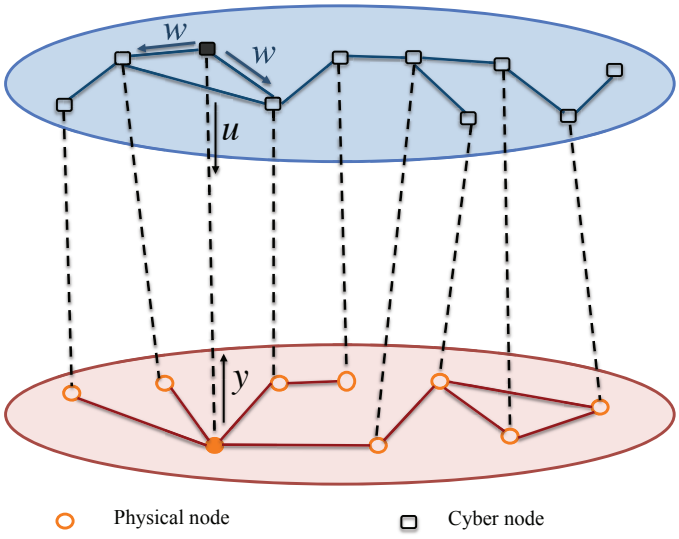


Figure 2.4: Illustration of "One-to-one" interdependent model.

Note that for $0 \leq \epsilon < 1$,

$$x_1 = \rho(\epsilon) = \sum_{i \geq 1} \rho_i \epsilon^i < \sum_{i \geq 1} \rho_i \epsilon = \epsilon. \tag{2.12}$$

Therefore, at the end of the first iteration, the probability of failure for a physical node decreases. The same analysis as above can be carried out for any iteration l. If we denote the physical node failures at the beginning of iterations l and $l+1$ by x_l and x_{l+1}, respectively, then

$$x_{l+1} < x_l, \tag{2.13}$$

which means that our simple (and somewhat intuitive) healing rule for the Buldyrev et al. network will always lead to its complete healing. This of course will not be the case for more complicated network models with complex contagion and healing rules. However, the message-passing approach used in this section can be generalized to develop a framework for studying such cases. The rest of this work is dedicated to this task. Section 2.4 sets up the

network model and formulates the message-passing problem for the general case. Section 2.5 then generalizes the technique used here by applying a *density evolution analysis* to study the dynamics of the cascade in the network.

2.4 Problem Formulation and Modeling

This section presents a graphical model for studying message passing in cyber-physical systems. First, we explain our network model for both physical and cyber networks. Then, we describe our models for the initial disturbance, healing, and contagion within each network and between the two networks. Finally, we explain how our modeling framework can be applied, in an abstract level, to a network of autonomous cars as an example of cyber-physical systems.

2.4.1 Network Model

For our analysis, we consider random networks with given degree distributions as models of cyber and physical networks. This enables us to model random networks with arbitrary degree distributions such as scale-free networks with a power law degree distribution [43], and Erdős-Rényi random graphs with a Bernoulli degree distribution [44]. We define cyber (physical) degree of a node as the number of nodes in the cyber (physical) network connected to the node. In a similar fashion to codes on graphs, we use polynomials to represent the degree distributions of the networks:

$$\rho(z) = \sum_{i \geq 1} \rho_i z^i, \text{and} \quad \lambda(z) = \sum_{i \geq 1} \lambda_i z^i \tag{2.14}$$

denote the degree distributions of the cyber and physical networks, respectively, where ρ_i is the fraction of cyber nodes with cyber degree i, and λ_i is the fraction of physical nodes with physical degree i.

To capture the interconnections between the two networks, two more polynomials are needed: one for the physical degree distribution of cyber nodes, and one for the cyber degree distribution of physical nodes. However, in order to simplify the presentation of results, we

assume that each cyber node controls a physical nodes, while each physical node is connected to one cyber node. The analysis for the general case of degree distributions could be carried out along the same lines as the analysis in this work.

2.4.2 Initial Disturbance, Contagion, and Healing

Here, we explain our models for the initial disturbance, contagion within each system and between the two, and healing of the physical system by the cyber system. Our methodology, however, could be extended to a wide range of models.

- **Initial disturbance**: We assume that each physical node initially fails with a small probability ϵ, where $\epsilon \ll 1$, independently from other nodes. In this work, we only consider initial disturbance for the physical network. The analysis for the case of a cyber attack could be conducted in a similar fashion.

- **Contagion within physical network**: After being defected, a physical node may defect each of its neighbors with probability p. This probabilistic model is commonly used in the literature for a range of applications [45].

- **Healing of physical nodes**: A cyber node heals a physical node if that physical node is its only defected physical neighbor. An example of this could be a control center that has all measurements but one from the power grid, so it must derive the phase or voltage value for the remaining component.

- **Contagion from physical to cyber system**: A cyber node with no functioning physical neighbor will go out of service. An example could be an internet server that looses its power supply in a power outage.

- **Contagion within cyber system**: If all cyber neighbors of a cyber node are out of service, then the cyber node itself will go out of service. An example could be an internet server whose neighboring servers have all been disconnected from the network.

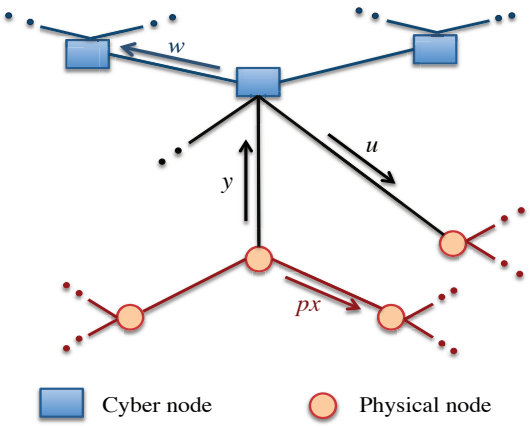


Figure 2.5: Example of messages exchanged in cyber-physical system.

2.4.3 Message Passing in Cyber-Physical Systems

In our model, the interactions between nodes are represented by messages. Accordingly, all sorts of contagions and the healing process scenarios explained above could be interpreted in a message-passing framework as follows:

1. Defection (D) message:

   - A defected physical node sends a defection message D to its cyber neighbors with the probability of y. It also sends a message D to each of its physical neighbors with probability p.
   - A defected cyber node sends a D-message to its cyber and physical neighbors with the probability of w.
   - A functioning cyber node that cannot heal a physical node sends a D-message to that node with the probability of w.

2. Healing (H) message: A cyber node that is able to heal a physical node sends a healing message H to that node.

Defining the messages as above, shown in Figure 2.5, addresses the first two differences listed in Section 2.2 between codes on graphs and CPSs. Note that these messages are introduced to

capture the interactions in the CPS, while they may not be actually exchanged between the nodes in the underlying networks. On the other hand, it is also possible that the messages do not arrive at the destinations due to imperfections of lines/channels in a real CPS. To address this point, we consider "missing probabilities" for the messages. The probability of missing messages exchanged within the physical network, within the cyber network, and between the physical and cyber networks is represented by P_{mp}, P_{mc}, and P_{mi}, respectively. The value of these parameters may vary between 0 and 1 according to the underlying application.

2.4.4 Autonomous Cars

Recent advances in signal processing, communications, network monitoring, and control systems pave the way to the next generation of vehicles, such as autonomous cars. Autonomous cars have attracted investments from many companies such as Wayco (formerly known as the Google self-driving car), motivated by improved pollution control, ease of use, and safety. A self-driving car requires five basic functions in order to drive autonomously: localization, perception, planning, vehicle control, and system management [46, 47]. The localization function finds the estimated position of the vehicle based on GPS, and the perception function obtains the information of the surrounding environment using car sensors. Through this gathered information, the planning function provides the maneuvers of the self-driving car. The vehicle control function applies commands of the planning function by accelerating, braking, and steering the car. Finally, the system management function provides the supervision of self-driving cars. While the first four functions operate the physical aspects of the driverless car, the system management function forms a new network layer, called the *vehicular cloud*, which maintains smooth traffic flow on roads through effective communication and distributed processing [48]. Hence, a CPS could be defined with the vehicular cloud as the cyber network and autonomous cars as the physical network. The message-passing framework introduced earlier in this section could then be applied to study the operation of this CPS as follows:

- **Initial disturbance**: Each autonomous car could fail due to any abnormality in the vehicle operation, from an engine problem to an unexpected speed or direction, as shown in Figure 2.6.

- **Contagion within physical network**: Failure in an autonomous car could cause problems for neighboring cars. The failure could be healed by the supervising cyber nodes or it could result in cascading failure in the network [49].

- **Healing of physical nodes**: Supervisors in the cyber network have access to essential data and services, from typical autonomous car measurements in the region to routes to service centers. The cyber node could use these services to help the autonomous car by the following: i) reporting a command to the planning function of the autonomous car in order to update the vehicle control or sending it to the nearest service center, or ii) providing an alarm to the passengers of the car to take appropriate action against the failure.

- **Contagion from physical to cyber system**: If some physical nodes send the wrong measurements of their status on roads, then the supervising cyber node may improperly perceive the situation.

- **Contagion within cyber system**: False information in a cyber node would lead to sending wrong information to other cyber nodes.

2.5 Density Evolution Analysis of Cyber-Physical Systems

In order to study the impact of an initial disruption, we keep track of D and H messages by employing density evolution. Recall the example of Figure 2.3 where variable and check nodes exchanged messages with values 0, 1, or ϵ in consecutive iterations. Density evolution tracks the density of D messages (e.g., ϵ) as the number of iterations grows. This density is defined as the fraction of D messages among all messages exchanged between the variable nodes and check nodes at each iteration. The message-passing algorithm of Section

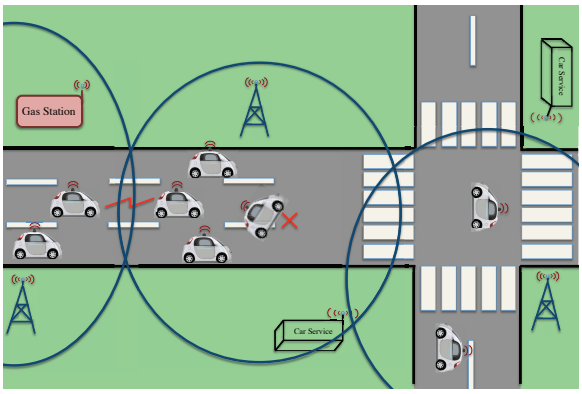


Figure 2.6: Illustration of autonomous cars' message- passing to avoid congestion and failure propagation.

2.2 is able to fix all damaged variable nodes, if and only if the density of D messages converges to 0 as the number of iterations grows. Employing the concept of density evolution used in codes on graphs, we obtain a DE equation for CPSs.

Theorem 1. *The density evolution equation for the system defined in Section 2.4 can be obtained as follows:*

$$\begin{aligned} x_0 &= \epsilon, \\ x_l &= f(x_{l-1}), \end{aligned} \tag{2.15}$$

where

$$f(x_{l-1}) = A \times B + A \times [1 - B] \times P_{mi}, \tag{2.16}$$

and

$$A = x_{l-1} + \Big(1 - x_{l-1}\Big)\Big(1 - \lambda(1 - px_{l-1})\Big)\Big(1 - P_{mp}\Big),$$

$$B = 1 - \Bigg\{\Bigg[\Bigg(1 - x_{l-1} - \Big(1 - x_{l-1}\Big)\Big(1 - \lambda(1 - px_{l-1})\Big)$$

$$\Big(1 - P_{mp}\Big)\Big)\Big(1 - P_{mi}\Big)\Bigg]^{a-1} \times \Bigg[1 - \rho\Big(\Big(\Big[1 - P_{mi}\Big]$$

$$\Big[x_{l-1}+\Big(1-x_{l-1}\Big)\Big(1-\lambda(1-px_{l-1})\Big)\Big(1-P_{mp}\Big)\Big]\Big)^{a}$$
$$\Big(1-P_{mc}\Big)\Big)\Big]\Big\},$$

and a is the number of physical nodes under each cyber node. The system heals if and only if $x_l \to 0$ as $l \to \infty$.

Proof. After a disturbance has occurred, D messages will be generated by the failed nodes. Since in our model each node is disturbed by a probability ϵ, the initial density of D's, denoted by x_0, is ϵ. Assume that we are at the beginning of the l-th iteration and find x_l in terms of x_{l-1}. In order to obtain the recursive equation during the $l-1$-th iteration, we need to define y, u, and w rigorously. To this end, we suppose that exchanged messages are independent of each other. The next theorem shows that this assumption holds if the number of nodes is sufficiently large in the network. According to the definition of failure messages from a physical node to a cyber node, and the probability of missing messages, y can be written as

$$y=\Pr\Bigg\{\Big(\text{Physical node has failed}\Big)\bigcup\Big(\text{Node has not failed}\bigcap\text{At least one of its physical neighbors has failed}\bigcap\text{message from the neighbor has not been missed}\Big)\Bigg\} \tag{2.17a}$$

$$\Rightarrow y=x_{l-1}+\Big(1-x_{l-1}\Big)\Big(1-\lambda(1-px_{l-1})\Big)\Big(1-P_{mp}\Big). \tag{2.17b}$$

Also, the failure message from a cyber node to a physical node, u, can be derived as

$$u=\Pr\Big\{\text{Cyber node received the failure of at least one of its physical nodes}\bigcap\text{Cyber node has access to at least one healthy cyber node}\Big\}, \tag{2.18a}$$

$$\Rightarrow u=1-\Bigg(\Big(1-y\Big)\Big(1-P_{mi}\Big)\Bigg)^{a-1}\Bigg(1-\rho\Big(w\,(1-P_{mc})\Big)\Bigg). \tag{2.18b}$$

Finally, the failure message between cyber nodes is defined as

$$w = \Pr\Big\{\text{Cyber node receives message of failure of all of its physical nodes}\Big\} \tag{2.19a}$$

$$\Rightarrow w = \Big(y\left(1 - P_{mi}\right)\Big)^{a}. \tag{2.19b}$$

At the l-th iteration, x_l can be defined as

$$x_l = \Pr\Bigg\{\Big(\text{Physical node fails} \bigcap \text{Its cyber node sends D message}\Big) \bigcup \Big(\text{Physical node fails} \bigcap \text{H message from its cyber node is missed}\Big)\Bigg\} \tag{2.20a}$$

$$\Rightarrow x_l = y \times u \; + \; y \times (1-u)\, P_{mi}. \tag{2.20b}$$

We can eliminate w by substituting (2.19b) into (2.18b) and obtaining u as a function of y. Substituting (2.17b) into (2.18b) and then (2.20b) yields (2.15). □

The above formulas were obtained with the assumption of independence between messages in the network. In the following theorem, we utilize the LDPC code analysis [40] to show the existence of such an independence between messages in a CPS.

Theorem 2. *For the cyber-physical system described in Section 2.4, if the number of nodes is sufficiently large, then the incoming messages to each cyber or physical node can be considered as independent messages.*

Proof. Let us divide all messages in a CPS into two parts: (a) messages between two networks, or inter-messages, and (b) messages within networks, or intra-messages. The edges and nodes for the inter-messages create the form of bipartite graphs. Richardson and Urbanke [40] showed that if the length of the smallest *cycle* in this bipartite graph is greater than $2l$, or in other words, if the neighborhood of length $2l$ for every node is *cycle-free*, then, up to the lth iteration, the incoming messages to each check node or variable node are inde-

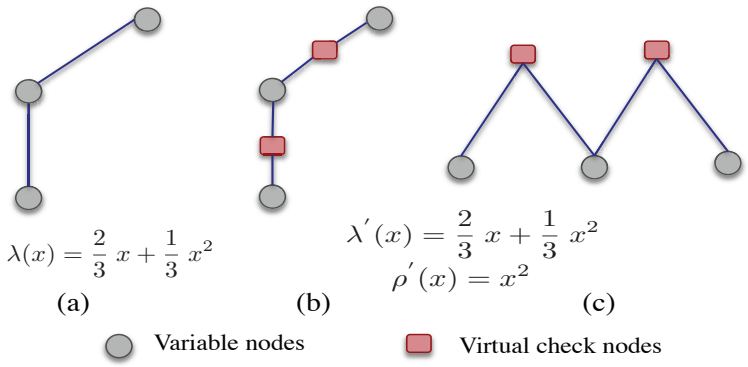


Figure 2.7: Example of bipartite graph with virtual check nodes: (a) variable nodes, (b) virtual check node inserted between every two variable nodes, and (c) variable nodes and virtual check nodes forming a bipartite graph.

pendent from each other. The authors then proved in Appendix A of [40] that, for any given l, such a cycle-free structure is achieved for sufficiently large number of nodes. The same exact proof applies to the bipartite graph connecting the cyber and physical networks as the number of nodes grows very large, thereby guaranteeing the independence of *inter*-messages.

There are no intra-messages in the message passing in LDPC codes as the Tanner graph is bipartite. The existence of intra-messages in CPSs is a direct consequence of the second difference between LDPC codes and CPSs listed in Section 2.2. To obtain the independence between intra-messages, we first map each node (physical or cyber) to a variable node as shown in Figure 2.7(a). Without loss of generality, we add a virtual check node between every two connected variable nodes as shown in Figure 2.7(b). This virtual check node acts as a relay for the messages and does not have any impact on the message passing. The resulting graph can now be simply considered as a bipartite graph as shown in Figure 2.7(c). As seen in the figure, the degree distribution of variable nodes remains the same while the degree distribution of virtual check nodes always equals to x^2. Obtaining such a bipartite graph enables us to apply the same logic of inter-messages using [40] for the intra-messages of a network. That is, if the number of nodes in a network (either physical or cyber) with a given degree distribution is sufficiently large, then the network satisfies the cycle-free condition as stated above, which guarantees the independence of *intra*-messages. □

2.5.1 Steady-State Behavior of Cyber-Physical Systems

We now study the steady-state behavior of the cyber-physical system against a failure for the defined message-passing rules. After a failure occurs in the network, defection messages appear in the network. It is expected that if the number of message-passing iterations increases, the density of defection messages through density evolution analysis approaches to 0 or 1, which means that the system reaches a steady-state condition that would be either complete healing or complete collapse. We prove this claim in the following theorem and confirm it via simulations in Section 4.6.

Theorem 3. *For the cyber-physical system defined in Section 2.4, if the message-passing iterations increase, then the system will reach a steady-state condition, which is a complete-healing state or a complete-failure state.*

Proof. We first show that the theorem holds for a simple network. We then extend the theorem truth to more general networks. To begin with, consider a network with one cyber node and two physical nodes, as shown in Figure 2.8(a). If μ_1 denotes the probability of failure for a physical node, then one of the following probabilities may occur:

$$\begin{cases} \binom{2}{0} \mu_1^0 \left(1-\mu_1\right)^2 & Case\ I: 0\ node\ failure, \\ \binom{2}{1} \mu_1 \left(1-\mu_1\right) & Case\ II: 1\ node\ failure, \\ \binom{2}{2} \mu_1^2 \left(1-\mu_1\right)^0 & Case\ III: 2\ nodes\ failure. \end{cases}$$

For cases I and III, the system is in a steady-state condition. In case I, none of the nodes is affected by the failure and the network is healthy. In case III, two nodes are lost due to failure. As we assumed (like LDPC codes), if more than one physical node is lost, then the corresponding cyber node cannot heal them. Hence, the physical nodes remain failed and in turn cause the cyber node to fail. Therefore, the network goes into complete-collapse. In

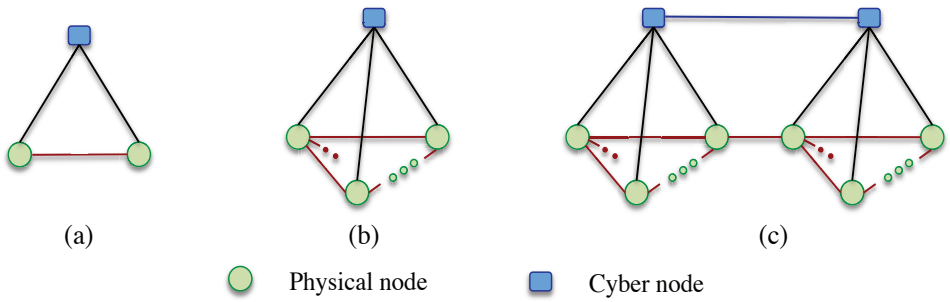


Figure 2.8: Illustration of cyber-physical system graph in proof of Theorem 3: (a) simple CPS, (b) CPS with one cyber and n physical nodes, and (c) CPS with two cyber nodes and n physical nodes.

case II, however, the system would be in a transient condition, which means that the network has neither completely healed nor completely failed. This occurs when a cyber node heals the failed node, but the failed node already propagates the failure to one of its neighbors. Hence, there is a failed node in the next state. After the l-th iteration, the probability of the network to be in a transient condition is

$$\binom{2}{1} \mu_1 \left(1-\mu_1\right) \left((1-p)^{(\alpha-1)} p\right)^l, \tag{2.21}$$

where p represents the probability of failure propagation between physical nodes, and α shows the number of neighbors for the failed node. For simplicity, we have assumed that all nodes have the same number of neighbors. As the l grows, the probability of a network to be in a transient condition goes to zero. Therefore, the system reaches a steady-state healing condition (case I + no failure propagation during message-passing in case II) or a steady-state collapsed condition (case III + at least two failed nodes due to failure propagation in case II).

The assumption of two physical nodes can be extended to n physical nodes, as shown in Figure 2.8(b). In the same fashion as above, the probability of transient condition in such networks after the l-th iteration would be

$$\binom{n}{1} \mu_1 \left(1-\mu_1\right)^{(n-1)} \left((1-p)^{(\alpha-1)} p\right)^l. \tag{2.22}$$

As $l \to \infty$, the probability of transient condition goes to zero.

We then increase one cyber node to m cyber nodes by defining clusters. A cluster includes a cyber node and its supporting physical nodes. Figure 2.8(c) shows two clusters ($m = 2$). By employing the above conclusion, the probability of one cluster with k physical nodes being in a transient condition is

$$\binom{k}{1} \mu_2 \left(1-\mu_2\right)^{(k-1)} \left((1-p)^{(\alpha-1)}\, p\right)^l, \tag{2.23}$$

where μ_2 is the probability of failure for a physical node in one cluster. Hence, for v out of m clusters with one failed physical node, the probability of a transient condition in entire network after the l-th iteration would be

$$\begin{aligned}\binom{k_1}{1} \mu_2 \left(1-\mu_2\right)^{(k-1)} \left((1-p)^{(\alpha-1)}\, p\right)^l, \times \ldots \times \\ \binom{k_v}{1} \mu_2 \left(1-\mu_2\right)^{(k_v-1)} \left((1-p)^{(\alpha-1)}\, p\right)^l,\end{aligned} \tag{2.24}$$

where k_i represents the number of nodes in the i-th cluster. As the number of iterations grows, the transient condition gradually vanishes. In other words, if during one of these iterations all nodes becomes healthy, then the network become healthy. Also, if two nodes in one cluster fails during the iterations, then the failures gradually permeate among the nodes in the cluster and then all nodes in the network. Therefore, a cyber-physical system with m cyber nodes and n physical nodes reaches a steady-state condition. □

2.5.2 Sufficient Condition for Healing

Once the recursive equation of density evolution is derived for a given set of contagion and healing rules, it can be utilized in many ways to gain useful insights into the network design. The following theorem, for example, employs equation (2.15), with the assumption of no missing messages ($P_{mi} = P_{mc} = P_{mp} = 0$), to obtain a sufficient condition for the system to heal completely.

Theorem 4. *The cyber-physical system described in Section 2.4 with degree distribution pair* (λ, ρ)*, and parameters* a *and* p *heals if*

$$x_0 < \frac{1}{\Big(a-1\Big)\Big(1+p\lambda'(1)\Big)^2} \tag{2.25}$$

Proof. By taking the Taylor series from the right side of (2.15) at $x_{l-1} = 0$, we obtain

$$\begin{aligned} x_l = &\Big(a-1\Big)\Big(1+p\lambda'(1)\Big)^2 x_{l-1}^2 - 0.5\Big(a-1\Big)\Big(1+p\lambda'(1)\Big) \\ &\times\Big[\Big(a-2\Big)\Big(1+p\lambda'(1)\Big) + 2p\Big(2\lambda'(1)+p\lambda''(1)\Big)\Big]\, x_{l-1}^3 \;+\; O(x_{l-1}^4). \end{aligned} \tag{2.26}$$

For x_l to be less than x_{l-1}, it is enough to show that x_{l-1} is larger than the first term on the right side of (2.26). That is,

$$\Big(a-1\Big)\Big(1+p\lambda'(1)\Big)^2 x_{l-1}^2 < x_{l-1}. \tag{2.27}$$

Inequality (2.27) holds for every l if it holds for $l = 1$. Substituting $l = 1$ in (2.27) leads to inequality (2.25). □

Theorem 4 provides some interesting intuitions. First, note that

$$\lambda'(1) = \sum_{i\geq 1} i\lambda_i \times x^{i-1}|_{x=1} = \sum_{i\geq 1} i\lambda_i, \tag{2.28}$$

is the average degree of the physical nodes. Theorem 4 indicates the necessity of a low average degree for the physical nodes for achieving a resilient system. This is because in our model, physical nodes with higher degrees can damage more nodes. Second, this theorem suggests that the number of physical nodes under each cyber node, a, should be kept small. This increases the chance of healing physical nodes since a cyber node needs to have all but one measurements to heal a physical node. Finally, the theorem implies that smaller values of p are desirable, which is expected.

2.5.3 Optimizing for Resiliency

We now study design implications of the density evolution analysis of the previous section. Based on the analysis done in Theorem 4, with the given network parameters λ, ρ, a, and p, one could evaluate the upper bound on x_{l-1}. Here, we refer to this value as ϵ_s. Also, one could employ the recursive equation of (2.15) to find the most severe disruption that can be tolerated by the network. To this end, we formulate an optimization problem with respect to network constraints. The solution to this problem would be the values of network parameters that achieve maximum resiliency against initial disturbances. We represent the maximum initial disturbance by ϵ_{max}. To obtain ϵ_{max} given a and p, an optimization problem can be set up as follows:

$$\begin{aligned} \underset{\{\rho_i,\lambda_i,\epsilon\}}{\text{argmax}} \quad & f(x_{l-1},\lambda_i,\rho_i) \\ \text{subject to} \quad & x_l = f(x_{l-1},\lambda_i,\rho_i), \\ & \sum_{i\geq 2}\lambda_i = 1, \\ & \sum_{i\geq 2}\rho_i = 1, \\ & 0 \leq \lambda_i \leq 1, \\ & 0 \leq \rho_i \leq 1. \end{aligned} \tag{2.29}$$

Sometimes, for simplicity of analysis, we assume that $\rho(x) = x^M$ for some $M \geq 2$. We numerically solve this optimization problem for two scenarios. In the first case, we fix the network parameters and find ϵ_{max}. In the second case, we run the program to reveal λ_is that give us the largest ϵ_{max}. We will comprehensively discuss these results in Section 4.6.

Nevertheless, it is worth noting that the application of our proposed message-passing framework is not limited to the particular setting explained in Section 2.4. This framework could be applied to any set of contagion models, healing rules, and network structures for which a density evolution analysis could be carried out. Also, this analysis holds for

delay-free CPSs where cyber nodes respond immediately to failures in the physical network. However, in practical cases, cyber nodes may need a while to process the messages, collect the information, and take action. The crucial role of this time delay in the healing process will be investigated in the next section.

2.5.4 Analysis of Message Passing with Time Delays in Cyber Nodes

We now develop the above analysis by considering processing time delay in a CPS. To this end, we employ the definition of time slots. Previously, we have assumed that each iteration of message passing can be completely done in one time slot. However, if a failure occurs for a physical node, then the corresponding cyber node usually needs a few time slots to respond to the D message. This delay would be for a number of reasons, such as recovering data from the database, collecting data from other physical nodes, gathering information from neighboring cyber nodes, etc. Therefore, one iteration would need a few time slots in order to be accomplished. Cyber nodes usually react against a failure in a few time slots. In what follows, we derive the density evolution equation assuming that each cyber node needs two time slots to respond to a failure. For the purpose of simplicity and without loss of generality, we assume that messages are delivered at the destination nodes (i.e., $P_{mi} = P_{mc} = P_{mp} = 0$). For brevity, we skip most details of the definitions and give the final equations that describe the messages at each time slot.

Theorem 5. *For the cyber-physical system defined in section 2.4 with a cyber-node processing delay of two time slots, a density evolution equation for the l-th iteration can be obtained as*

$$\begin{aligned} x_l(t+3) &= f(x_{l-1}(t)), \\ f(x_{l-1}(t)) &= A \times B + C \times \big[1 - B\big], \end{aligned} \tag{2.30}$$

where A, B, and C are given as

$$A = \left\{ \lambda \left[1 - p \left(\lambda \Big(1 - p x_{l-1}(t) \Big) \times \Big(x_{l-1}(t) - 1 \Big) + 1 \right) \right] \right\}$$

$$\times \Big\{\lambda\Big(1-px_{l-1}(t)\Big)\times\Big(x_{l-1}(t)-1\Big)\Big\}+1,$$

$$B = \; 1-\Big\{\lambda\Big(1-px_{l-1}(t)\Big)\times\Big(x_{l-1}(t)-1\Big)\Big\}^{(a-1)}\times$$
$$\Big\{1-\rho\Big(x^a_{l-1}(t)\Big)\Big\},$$

$$C = \; 1-\lambda\Big(1-p\;A\Big).$$

The system heals if and only if $x_l(t+3)\to 0$ *as* $l\to\infty$ *(or equivalently,* $t\to\infty$*).*

Proof. Let $x(t), y(t), u(t)$ and $w(t)$ denote the messages in the t-th time slot (see Figure 2.5). If the probability of failure at the beginning of the t-th time slot is ϵ, then we have $x(t)=\epsilon$. In the next time slot, according to the model described in Sections 2.4.2 and 2.4.3, we have

$$y(t+1) = \; x(t)+\Big[1-x(t)\Big]\;\Big[1-\lambda\Big(1-px(t)\Big)\Big], \quad w(t+1) = \; \Big(y(t)\Big)^a,$$
$$u(t+1) = \; 1-\Big[1-y(t)\Big]^{(a-1)}\Big[1-\rho\Big(w(t)\Big)\Big], \quad x(t+1) = \; y(t+1), \tag{2.31}$$

where $y(t)=x(t)$, and $w(t) = \; \big(x(t)\big)^a=\epsilon^{\,a}$. Similarly, in the $(t+2)$-th time slot, we obtain

$$y(t+2)=x(t+1)+\Big[1-x(t+1)\Big]\;\times\;\Big[1-\lambda\Big(1-px(t+1)\Big)\Big],\; w(t+2)=\Big(y(t+1)\Big)^a,$$
$$u(t+2)=1-\Big[1-y(t+1)\Big]^{(a-1)}\Big[1-\rho\Big(w(t+1)\Big)\Big],\; x(t+2)=y(t+2). \tag{2.32}$$

After two time slots for processing the data in a cyber node, the probability of failure of a physical node can be calculated as

$$x_l(t+3)=y(t+2)\;u(t+2)\;+\;\Big[1-\lambda\Big(1-px(t+2)\Big)\Big]\Big[1-u(t+2)\Big]. \tag{2.33}$$

If we substitute equation (2.31) into (2.32) and then the result into (2.33), equation (2.30) will be obtained. □

Extending the DE analysis to cyber node delays of more than two time slots can be done along the same lines as above. In the next section, through numerical results, we will fully study the impact of different processing time delays on a self-healing cyber-physical network.

2.6 Simulation Results

To make more concrete sense of the above analysis, we have numerically simulated the message passing over cyber-physical networks. First, we simulate the network without considering processing-time delay to find the role of each network parameter. Here, we referred to this simulation as scenario I. Next, we will investigate the performance of the self-healing method in the presence of a processing-time delay. This simulation creates scenario II. Finally, we will compare both scenarios to provide clear criteria for choosing network parameters for the desired resiliency of a CPS.

2.6.1 Numerical Results for Scenario I

This section provides a number of evaluations for the self-healing method without considering a processing-time delay. To begin with, Figure 2.9 shows the fraction of physical node failures for different numbers of iterations, l. As can be seen, if $l \to \infty$, then the function goes to a step function. This implies two steady-state conditions, which would be either a complete healing or a complete failure scenario in the network. These steady-state conditions confirm Theorem 3 in Section 2.5.

Figure 2.10 shows the fraction of physical nodes failure in a steady-state condition, x_∞, after an initial disturbance, for two popular network models: Erdös-Rényi (ER) networks and scale-free (SF) networks. For this simulation, we assumed that: i) both networks have the same average degree, which is equal to 1.4, and ii) both networks have the same minimum and maximum degrees (k_{min} and k_{max}) that can be obtained via $k_{max} = k_{min}N^{(\frac{1}{\gamma-1})}$ [50] in which $k_{min} = 1$, $N = 100$, and $\gamma = 2.8$ (for SF networks). Based on these assumptions, we ran the simulations and found that SF networks have more tolerance against an initial

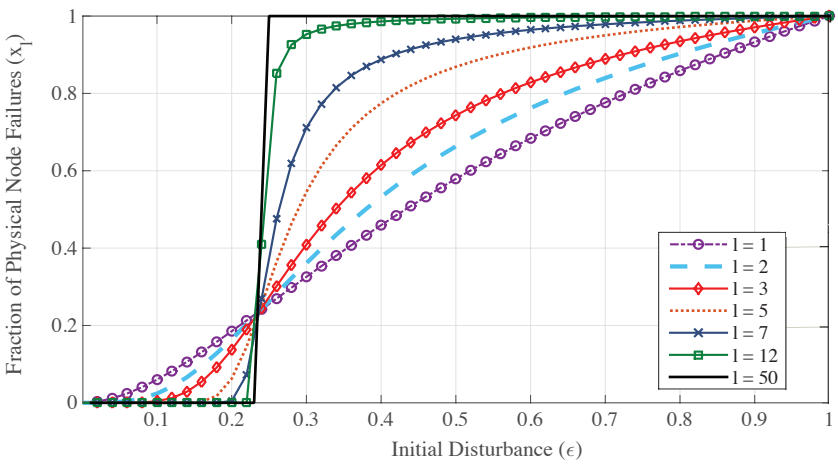


Figure 2.9: Probability of failure for physical nodes in different iterations w.r.t the initial disturbance, with network parameters $a = 5$, $p = 0.2$,$\lambda(z) = z^2$ and $\rho(z) = z^3$.

perturbation in comparison to ER networks, as shown in Figure 2.10. The reason could be that, given the same average degree, most nodes in a SF network have a lower degree than the nodes in an ER network. Therefore, given that the initial failures are selected randomly and unbiased, such nodes are less likely to be among the hubs in a SF network. This means less number of high-degree failed nodes, hence, less chance of failure propagation in the network. This finding confirms the results of Schneider et al. [22] stating that high-betweenness nodes should be planned as autonomous nodes, in order to have the best resiliency in an interdependent network.

In order to study the impact of missing messages on the resiliency of a self-healing interdependent network, we simulated our findings for a CPS with different values of P_{mi}, P_{mp}, and P_{mc}, and the following network parameters: $a = 4$, $p = 0.1$, and $\lambda(z) = \rho(z) = 0.5\,z + 0.4\,z^2 + 0.1\,z^3$. The results are shown in Figure 2.11. As can be seen in Figure 2.11(a), without missing messages between the two networks (i.e., $P_{mi} = 0$), the system is able to tolerate up to 55% initial loss of physical nodes. However, if P_{mi} gradually increases, then the opportunity of receiving H messages at physical nodes from cyber nodes continually decreases. Accordingly, the resiliency of the network is drastically reduced. This trend continues until the system completely loses its ability to heal a failure ($P_{mi} \geqslant 0.4$ for the

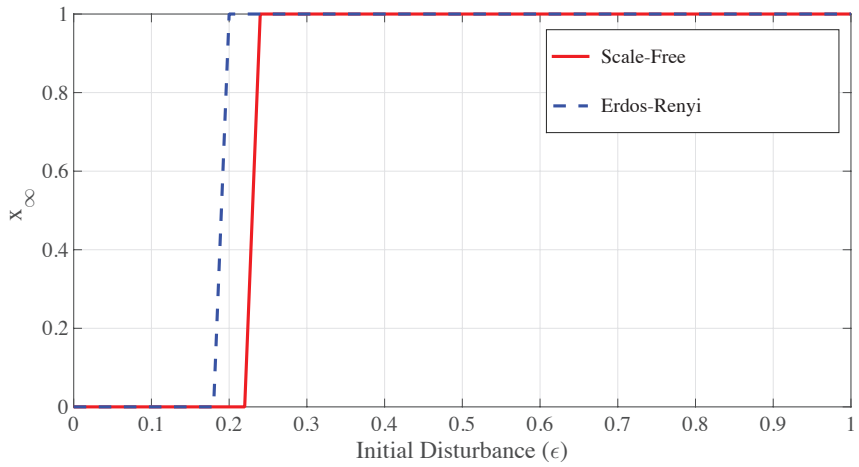


Figure 2.10: Steady-state fraction of physical failed nodes against an initial disturbance for Erdös - Rényi (ER) and Scale-Free (SF) networks. The average degree of networks is 1.4 with the min. degree of 1 and max. degree of 13, and $\gamma = 2.8$.

assumed network). Figure 2.11(b) shows that the increase in missing messages in the physical network (i.e., P_{mp}) improves the probability of healing in the system. However, this result is not surprising because a larger number of missing failure messages in the physical network means a smaller chance of failure propagation in this network. In our simulations, we also noted that changes in P_{mc}, the probability of a missing message within the cyber network, do not affect the network resiliency significantly, since the cyber nodes in our model mainly rely on information from their own physical nodes for their operation.

Figure 2.12(a) indicates the number of iterations needed for the network to reach a steady-state condition. As can be seen in Figure 2.12(a), the steady-state conditions would change from a complete healing (collapse) to a complete collapse (healing) for a *threshold* value of initial disturbance (ϵ), in this case $\epsilon = 0.23$. This threshold would be varied for different sets of network parameters. The more we move away from the threshold, the less iterations are needed for the network to reach its stable state. This is due to the limited capability of cyber nodes in healing the physical nodes. For example, for small disturbance such as $\epsilon = 0.05$, cyber nodes need only a few iterations to heal the network because almost all physical nodes are healthy. For larger disturbances, the chance of complete healing drops rapidly. Figure 2.12(b) shows the threshold that can be tolerated for a set of different physical

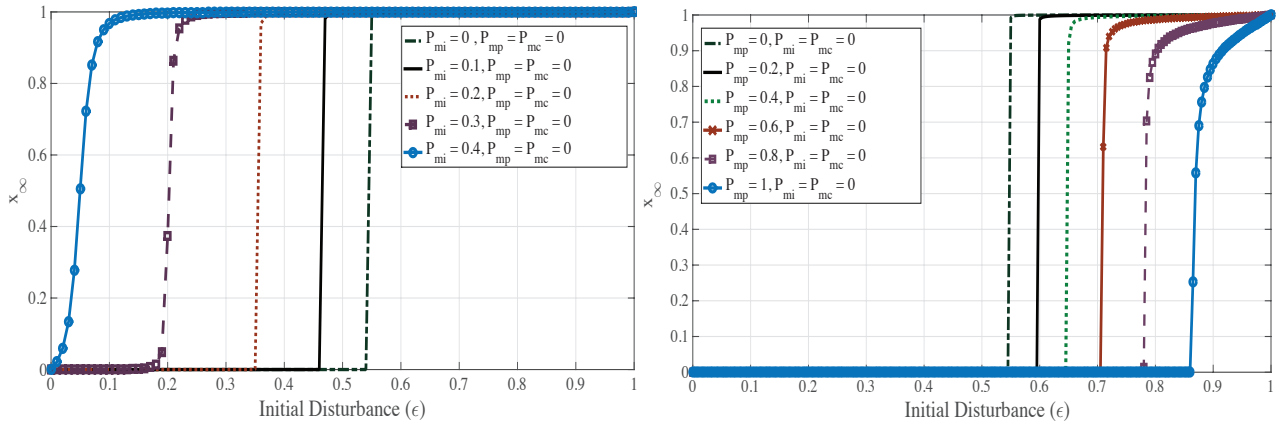


Figure 2.11: Influence of missing messages on probability of nodes' failure in CPS with following parameters: $a = 4,\ p = 0.1, and\ \lambda(z) = \rho(z) = 0.5\,z + 0.4\,z^2 + 0.1\,z^3$: (a) impact of missing messages between physical network and cyber network, P_{mi}, and (b) effect of missing messages in physical network, P_{mp}.

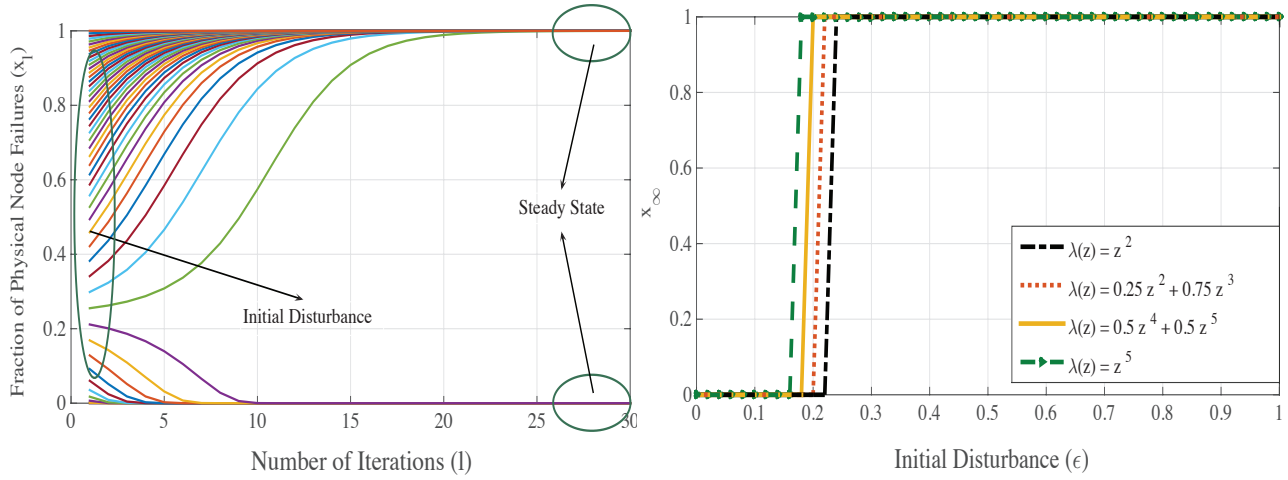


Figure 2.12: For network parameters $a = 5,\ p = 0.2$ and $\rho(z) = z^3$: (a) demonstrates the number of iterations needed for a network to be completely healed or failed for $\lambda(z) = z^3$ and (b) shows the probability of failure for physical nodes with different $\lambda(z)$ against an initial disturbance.

degree distributions in the network, while a and p are fixed. Also, we assume that enough message-passing iterations have already been done for the network to reach the steady state. As can be seen, the lower physical degree distributions have a higher resiliency against failure.

In order to find the maximum threshold and network parameters achieving this threshold, we numerically solve the optimization problem of (2.29). The problem is solved for two

Table 2.1: ϵ_s and ϵ_{max} for different network parameters and severity of initial disturbance

Table I.A: ϵ_s and ϵ_{max} for variation of a

p	a	$\lambda(z)$	$\rho(z)$	ϵ_s	ϵ_{max}
0.8	3	z^2	z^3	0.0740	0.1002
	5			0.0369	0.0482
	8			0.0211	0.0271

Table I.B: ϵ_s and ϵ_{max} for variation of p

p	a	$\lambda(z)$	$\rho(z)$	ϵ_s	ϵ_{max}
0.4	4	z^2	z^3	0.1028	0.1621
0.6				0.0688	0.0973
0.8				0.0493	0.0650

Table I.C: ϵ_s and ϵ_{max} for variation of $\lambda(x)$

p	a	$\lambda(z)$	$\rho(z)$	ϵ_s	ϵ_{max}
0.5	3	z^2	z^3	0.1250	0.1933
		z^5		0.0408	0.0525
		z^8		0.0200	0.0250

cases: (a) a, $\lambda(\cdot)$, $\rho(\cdot)$, and p are kept fixed and the maximum tolerated initial disturbance, ϵ_{max}, is found, and (b) a, p, and $\rho(\cdot)$ are fixed while λ_is are obtained for the ϵ_{max}. Results for the first and the second cases are shown in Table I and II, respectively, where the corresponding value of ϵ_s, the value of the upper bound in (2.25), is also listed for each set of network parameters. The following observations can be made from Table I:

- The results in Table I.A indicate that increasing a reduces the values of ϵ_s and ϵ_{max}. In fact, a large a increases the chance of receiving D messages by a cyber node from its physical neighbors. This reduces the resiliency of the system.

- The results in Table I.B confirm that reducing p leads to less vulnerability of physical nodes from their physical neighbors.

- Table I.C shows that a less-connected physical network results in larger values of ϵ_s and ϵ_{max} and, hence, higher resiliency.

Table 2.2: ϵ_s and ϵ_{max} for different degree coefficients of physical nodes.

p	a	$\rho(z)$	$\lambda(z)$	ϵ_s	ϵ_{max}
0.5	4	z^3	$\lambda_2 = 1$ $\lambda_3 = 0$ $\lambda_4 = 0$ $\lambda_5 = 0$	0.05334	0.07245
0.2	3	z^2	$\lambda_2 = 1$ $\lambda_3 = 0$ $\lambda_4 = 0$ $\lambda_5 = 0$	0.19531	0.35424

In the second case, we consider a physical degree distribution of minimum degree two and maximum degree five:

$$\lambda(z) = \lambda_2\, z^2 \;+\; \lambda_3\, z^3 \;+\; \lambda_4\, z^4 + \;\lambda_5\, z^5. \tag{2.34}$$

The outcomes for this case are shown in Table 2.2. As can be seen, we consistently obtain $\lambda_2 = 1$, while $\lambda_3 = 0$, $\lambda_4 = 0$, and $\lambda_5 = 0$. This is not surprising, because physical nodes with more physical neighbors help spread the failures. In fact, when solving the optimization equation (2.29) for scenarios with other limitations on $\lambda(z)$, we found that the degree of the physical nodes should be kept as small as possible.

2.6.2 Numerical Results for Scenario II

In Section 2.5.4, we have analytically obtained the influence of processing-time delay on the probability of failure in the CPS. Now, we numerically evaluate those results in the same line of scenario I. We begin with a processing-delay of three time slots. Hence, each iteration can be accomplished in four time slots. Figure 2.13(a) shows the number of time slots needed for the network to reach a steady-state condition. As can be seen, jagged lines occur as a result of the processing-time delay. Considering this delay, a cyber node needs k time slots (here, $k = 3$) to respond to the failure. During this time interval, the failure propagates in the physical network and could constantly increase the probability of failure for physical nodes. After the k-th time slot, however, the network would experience three

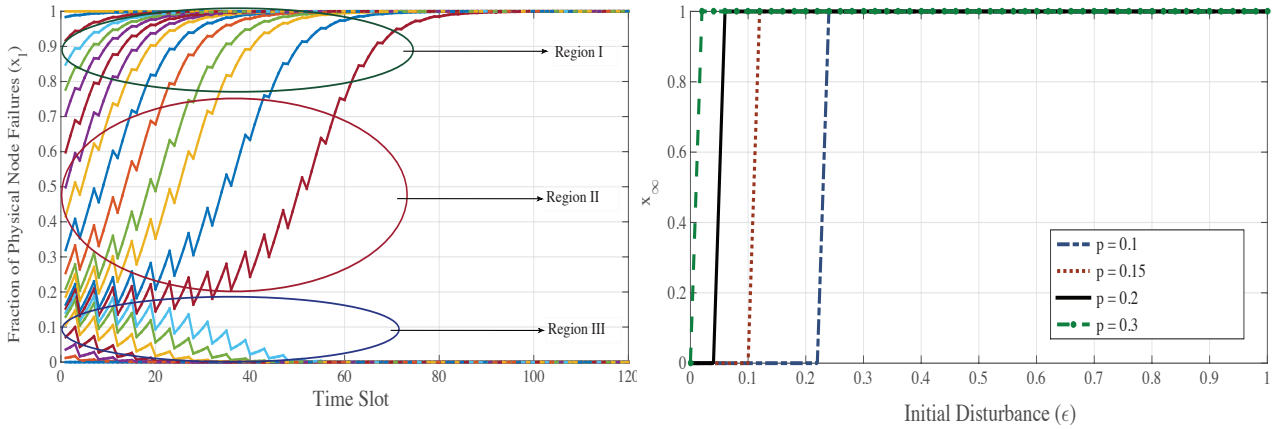


Figure 2.13: (a) The probability of failure for physical nodes in presence of processing-time delay. Processing delay = 3 time slots, and network parameters are $a = 5$, $p = 0.2, \lambda(z) = z^2$, and $\rho(z) = z^3$. (b) Impact of p on steady-state behavior of network.

cases for the probability of failure, depending on the ability of cyber nodes to heal their associated physical nodes:

- Region I: The healing ability of cyber nodes is diminishing due to the large number of failures in the network; hence, the probability of physical failures only increases.
- Region II: Cyber nodes are still able to heal their physical nodes, but this is not enough to stop the propagation of failure and change the overall trend. That is, the number of failures at the end of the iteration is still higher than that of the beginning of the iteration.
- Region III: The healing ability of cyber nodes outweighs the propagation of failure, leading to complete healing of the network after a few iterations.

The probability that each physical node gets affected by the failure of its physical neighbors, p, is decisive to the resiliency of a CPS. One example of this parameter in power systems could be revealed in protective relays. The mission of protective relays is to sense a fault and initiate a trip, disconnection or order. Therefore, good relays result in lower probability of failure in a network. The performance of relays, in a very abstract sense, can

be mapped to p. The influence of p becomes more crucial, when there is a processing-time delay for cyber nodes. Figure 2.13(b) shows the steady-state behavior of the network for different values of p, when the processing-time slot is three ($k = 3$). As can be seen, the higher value of p dramatically increases the vulnerability of the network against an initial disturbance. For instance, for $p = 0.3$, network resiliency is almost non-existent against any initial disturbance in the physical network.

2.6.3 Comparison of Results for Scenario I and II

Now, we compare scenarios I and II to understand the effect of delay on the performance of cyber-physical systems. First, consider Figure 2.14(a), which shows the direct impact of delay on maximum resiliency of a network. Delayed systems need considerably more time slots to be healed in comparison to non-delayed systems. This can be observed for $\epsilon = 0.1$. In addition, at $\epsilon = 0.3$, the non-delayed system can be cured after nine time slots. However, the delayed system completely collapses. The reason is that the failure propagates throughout the physical network during the three time slot delay interval. Figure 2.14(b) displays the steady-state behavior for delayed and non-delayed systems w.r.t. the size of the initial disturbance. A comparison of resiliency thresholds effectively demonstrates the essential need for a quick response from cyber nodes.

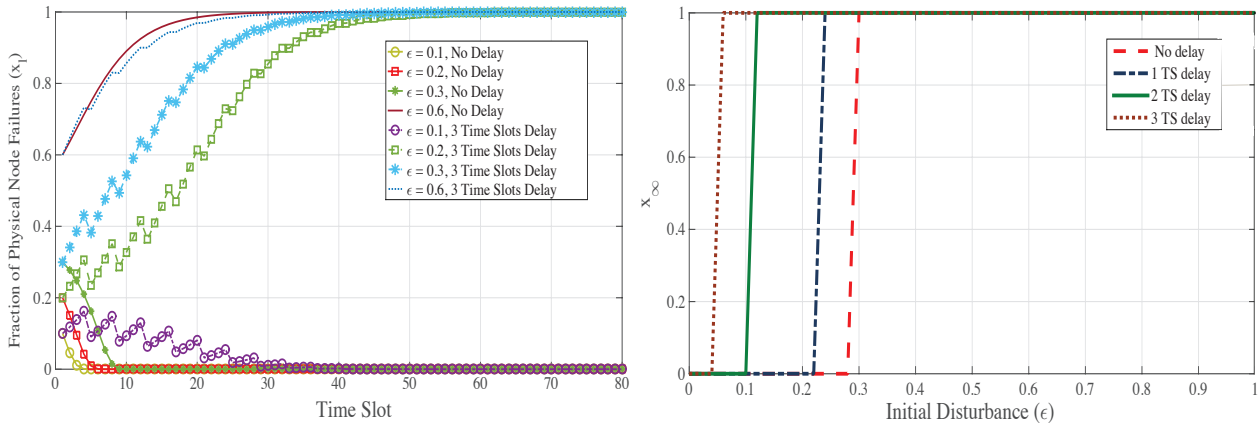


Figure 2.14: Comparison between non-delayed systems and delayed systems for network parameters $a = 5$, $p = 0.15, \lambda(z) = z^2$, and $\rho(z) = z^3$: (a) effect of time slot processing delay on systems, and (b) impact of time slot (TS) delays on maximum tolerated threshold in network.

2.7 Summary

We introduced a graphical model representation of cyber-physical systems and applied message passing to investigate the resiliency of inter-dependent CPSs. We provided a density evolution analysis to study the behavior of a system in the presence of both self-healing and propagation of failures. Our analysis resulted in a sufficient condition on choosing the network parameters for the system to completely heal after an initial disturbance. Then, we studied the steady-state behavior of cyber-physical networks after an initial disturbance, proving that the network reaches one of the two conditions, either a complete healing or a complete failure. To improve the network robustness, we set up an optimization problem to calculate network parameters for highest network resiliency against physical node disruptions, where we found the most severe attack that can be tolerated by the network, given a set of parameters. Finally, we studied self-healing cyber-physical networks where the response of cyber nodes to failures in the physical network is delayed. Our findings revealed the crucial importance of low processing-time delay for increasing the probability of healing in the network.

CHAPTER 3

RISK ASSESSMENT OF AUTONOMOUS VEHICLES USING BAYESIAN DEFENSE GRAPHS

Recent developments have made autonomous vehicles closer to being seen on roads. However, their security is still a major concern among drivers as well as manufacturers. Although some work has been done to identify threats and possible solutions, a theoretical framework is needed to measure the security of AVs. In this chapter, a simple security model based on defense graphs is proposed to quantitatively assess the likelihood of threats on components of an AV in the presence of available countermeasures. A Bayesian network analysis is then applied to obtain the associated security risk. In a case study, the model and the analysis are studied for GPS spoofing attacks, in order to demonstrate the effectiveness of the proposed approach for a highly vulnerable component.

The rest of this chapter is organized as follows. Section 3.1 provides an overview of threats against autonomous vehicles. Section 3.2 studies the use of Bayesian networks for security measurements. Section 3.3 briefly explains six major anti-spoofing techniques that are commonly used to protect GPS signals. Section 3.4 describes the proposed model, threat identification and risk assessment, and BN model in the presence of uncertainties in forming a defense graph. Section 3.5 applies the proposed model to the GPS unit as a highly vulnerable component in AVs. Various combinations of GPS anti-spoofing techniques are considered in measuring the protection levels provided by them collectively. Section 3.6 summarizes the chapter.

3.1 Review of Threats against Autonomous Vehicles

Threats to autonomous vehicles could be separately studied for autonomous automated vehicles (AAVs) and autonomous cooperative vehicles (ACVs). In the AAV case, a cyberattack might directly target some sources of information that a vehicle employs to

make a driving decision. Here, the internal processing unit plays a vital role in considering all available sources to determine the true state of the vehicle. The ability of the internal processing unit to identify threats and take countermeasures depends on the quantity and quality of techniques that are used. These techniques could be methods to anticipate pre-defined attack scenarios or the use of uncorrupted information to retrieve the true state. In the ACV case, in addition to AAV threats, communication among neighboring vehicles could provide additional information to verify the right decision for a vehicle. However, this data could potentially add some suspicious information at the same time. Therefore, communication and cooperation between vehicles provide a wider range of pros and cons that need to be considered in order to maximize the safety of vehicles. Figure 3.1 clearly shows both threats against AAVs and ACVs. Here, we provide an analytical platform using a Bayesian defense graph to evaluate threats in AAVs. This method is explained in section 3.2. On the other hand, threats against ACVs are studied via game theory, which captures the interactions between vehicles. A brief overview of game theory for vehicular networks is provided later on in section 4.1.

In general, the architecture of a smart vehicle can be assumed as a context-aware system that includes three fundamental parts: sensing and data collection, reasoning and processing, and application and communication, as shown in Figure 3.2(a). In the data collection part, devices are responsible for acquiring information from both external and internal sources. External sources refer to receiving messages from the outside of AVs such as neighboring cars and road signs. Internal sources refer to sensors and cameras in a vehicle that gather data from static and moving objects from the surrounding environment. In the reasoning and processing part, the fundamental aim is to determine the true state of the vehicle using certain and uncertain information. Certain information is the data that one could directly use to perform a specific task, whereas uncertain information is the kind of data that should be processed to make an inference. Hence, a processor is needed to apply the reasoning and algorithms. This part also includes powering, controlling, and database units

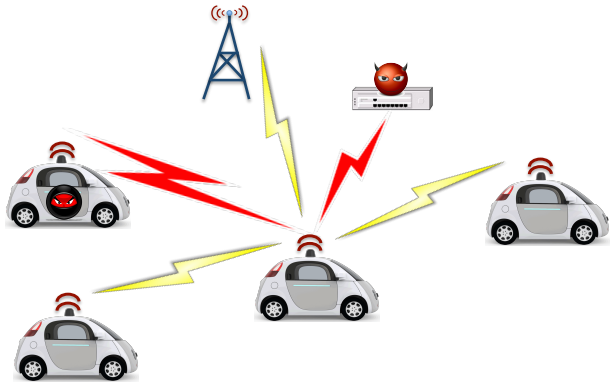


Figure 3.1: Threats for autonomous cars.

that are responsible for supplying, supervising, and storing predefined data, respectively. In the application part of the architecture, all decisions and instructions of an AV that are deduced from reasoning part will be implemented by sending regular and warning messages to the AV and its neighboring vehicles.

Since safety is among the highest priorities for an AV, careful attention must be devoted to the modeling of secure AVs. To this end, as can be seen in Figure 3.2(a), additional units using security equipment in the collecting data part and the security monitoring in the reasoning and processing part are considered. The security equipment could be either setting up extra secure techniques on existing components, e.g., arrival time techniques for a GPS component, or setting up some external hardware, e.g., redundant cameras. However, the security monitoring should be considered as a separate essential unit to process all required data for checking the safety of a vehicle. Figure 3.2(b) shows a typical security monitoring unit consisting of all major attack surfaces [6]. Here, each attack surface is referred to as a component. This unit as a part of processor has access to all required data for protection purposes. For instance, to protect GPS signals, the processor might need to use data regarding the security equipment unit, maps in the database, or even cameras, to ensure the dependability of receiving GPS signals.

In order to measure the vulnerability of each component shown in Figure 3.2(b), there exist some criteria that categorize threats as high, medium, and low. These criteria

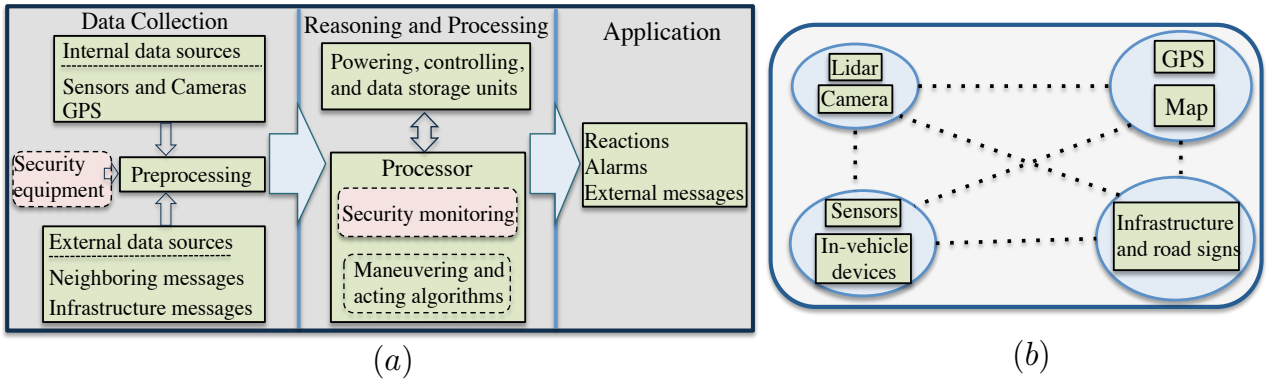
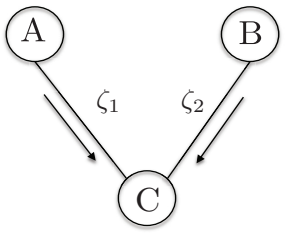


Figure 3.2: Secure autonomous vehicle: (a) architecture, and (b) security monitoring center.

are based on the feasibility of attack, the probability of attack success, the consequences of an exploited component for a vehicle, etc [9, 6]. Here, inspired by Bayesian inference over attack graphs, the level of security for an AV is quantitatively measured. Section 3.2 explains how attack graphs can be deployed to measure the security of networks.

3.2 Bayesian Networks for Security Measurements

There has been considerable amount of research regarding the development of Bayesian networks to measuring security in computer networks [51, 52, 53, 10, 54, 55, 56, 57]. The main motivations behind using BNs for security measurements are the following: (i) BNs provide a model of cause-effect relationships between different components of a system. The attack on each component can be considered a random variable that takes values from different domains. The relationship between components are captured by edges in a Bayesian graph that will show conditional dependency between nodes. Hence, the path (i.e., edges on a BN) that an attacker can move on and propagate its destructive behavior is determined. (ii) BNs provide a formulation for reasoning about partial beliefs under uncertainty. In particular, uncertainties can be assigned to the nodes and edges of a BN to evaluate the probability of successful attack on a particular component. (iii) BNs provide a platform to infer a belief w.r.t. different evaluations that come from various standard scoring systems and hypotheses. Specifically, depending on training data and an application, BNs are able



(a)

A	B	C	
		T	F
D	D	θ_1	$1-\theta_1$
	ND	θ_2	$1-\theta_2$
ND	D	θ_3	$1-\theta_3$
	ND	θ_4	$1-\theta_4$

(b)

Figure 3.3: (a) Portion of Bayesian network, and (b) corresponding conditional probability table.

to learn different values (weights) for their nodes, edges, and hence achieve a conclusion accordingly.

In order to assess a BN for security measurements, a simple case is explained to show the logic of Bayesian inference over a predefined graph. One can find comprehensive details in the mentioned literature. Bayesian graphs are usually defined as directed acyclic graphs (DAGs), where nodes are connected by directed edges and each node, except root nodes, has parent nodes. Figure 3.3 illustrates a simple example of a BN, where dependency between nodes are illustrated in the conditional probability table (CPT). In this figure, A and B are two parents for C. Also, assuming there is a potential attack, D and ND imply detection and node detection, respectively, of any malicious activities. In addition, T and F imply that the attack has been detected or not detected, respectively. Here, the logic of detection between components is AND, meaning that C cannot detect any abnormalities if neither A nor B detects any abnormalities. Under these assumptions, the vulnerability of C can be obtained as

$$p(C) = \sum p(C|A,B)\ p(A,B), \tag{3.1}$$

$$p(C|A,B) \propto p(A,B|C)p(C). \tag{3.2}$$

Equation (3.1) is a marginal probability distribution to obtain prior probability for C, and equation (3.2) represents a posterior probability distribution using the prior probability and

$p(A, B|C)$ as a likelihood distribution. Using both equations, it is possible to calculate the likelihood of a successful attack on C, given the vulnerability of A and B (C's parent nodes).

To assign the probability of a successful exploit for an element in BN (e.g., A in Figure 3.3), the following approaches can be used:

- Probabilities could be deduced from a large amount of training data. This data can be modeled through statistical approaches to obtain the highest possible expected value for each case.

- There are some standardized systems for scoring vulnerability, for example, the Common Vulnerability Scoring System (CVSS) or E-safety Vehicle Intrusion proTected Applications (EVITA). CVSS uses existing databases such as the National Vulnerability Database (NVD) to evaluate the vulnerability of an element. EVITA is a project co-funded by the European Union within the Seventh Framework Program for research and technological development.

- The probability for successfully exploiting a vulnerability could be inferred from several studies and papers that have already addressed similar issues.

In this work, the last two items will be applied, since it was too difficult to acquire a huge amount of data for the application here. Hereafter, the concept of CVSS as a standard scoring is briefly explained, and the others can be found in the literature. CVSS mainly uses three concepts to measure the vulnerability of a component: base score, temporal score, and environmental score. The base score quantifies the intrinsic attribute of each vulnerability, which is independent of time and user environments. The temporal score, however, assesses a vulnerability based on properties that might change over time. The environmental score considers the characteristics of a vulnerability that are relevant and unique to a particular application. Using these scores, the CVSS generates a value from 0 to 10 that is simply converted to a probability by dividing the score over 10. This number provides a probability

for an individual vulnerability that ignores causal relationships between exploits. Previous studies can be used to consider all dependencies between exploits (edges weights).

3.3 GPS Anti-Spoofing Techniques

The essential importance of secure GPS signals has initiated considerable research relative to GPS spoofing countermeasures. This section provides a brief overview of six major anti-spoofing techniques that are commonly used. By understanding each technique, an attack graph based on the relationships between elements of a technique can be built in order to assess the vulnerability of a GPS component.

3.3.1 Authentication

Authentication as an anti-spoofing technique has long been used in the military version of GPS receivers. However, its high reliability and low additional hardware requirements have made this method a potential candidate to protect current civil GPS signals [58, 59]. In this regard, some schemes have been proposed that are mostly based on public-key digital signatures such as the navigation message authentication (NMA) and spread spectrum security codes such as the public spreading code authentication (PubSCA). The process of authentication, however, takes some time to complete. However, this inevitable delay in many cases cannot meet the time-to-alert (TTA) requirements for ensuring integrity and safety of global navigation satellite system (GNSS) signals in civil aviation. Hence, while authentication techniques are relatively safe, they can be overcome by some spoofing methods that are complex enough. Therefore, using other anti-spoofing techniques will be a complement to authentication methods.

3.3.2 Timing Check

Spoofing transmitters and simulators are faced with an unavoidable delay in order to mislead GPS receivers. The delay might come from the required data processing by a spoofer, or propagation time between the spoofer and the target. In either case, the inevitable delay can be exploited as an opportunity to reveal the existence of a spoofer using accurate clock

consistency measurements or applying some techniques such as the time differential of arrival (TDOA), time of flight (ToF), etc [60, 61, 62]. Despite these methods, the ToA encounters some inherent limitations to detect counterfeit signals. For instance, the GPS data frame structure includes different parts and update frequencies that are widely known. Since update frequencies of most parts are slow, one could predict GPS data bits and transmit fake GPS signals.

3.3.3 Spatial Processing Using Multi-Antenna

While GPS receivers collect authentic signals from different satellites with different movement trajectories, fraud signals are usually transmitted from the same antenna due to logistical limitations. This creates an opportunity to employ a spatial processing technique to discriminate highly correlated signals from other probable (fake) signals. The spatial processing technique can be applied through antenna array structure methods. A multi-antenna receiver could be used to shape its beam to detect the direction of spoofing signal. Then, the receiver would be able to steer a null toward the direction of the fake signal to suppress it. Also, a synthetic antenna array could be employed using only one antenna to reduce the hardware complexity. This could be possible if one antenna is moved along a random trajectory and different pseudo-random noise (PRN) numbers are recorded for the GPS signal. In general, there are multiple methods for spatial processing, and each of them could be selected based on a related application [63, 64, 65].

3.3.4 Amplitude/Power Monitoring

Investigating the inherent received power from satellites and fraud transmitters reveals the existence of some differences that help us to distinguish between an authenticated signal and a counterfeit signal. Notable differences arise from the variation of received signal-to-noise ratio (SNR) power, and the strength of an absolute power. If a spoofer successfully mounts its attack on a GPS receiver, the received SNR might experience a sudden change that indicates the presence of counterfeit signals. On the other hand, it is difficult for a spoofer to estimate an accurate power to impose sufficient signal strength at the GPS re-

ceiver. However, despite the positive points of power-monitoring methods, they have their own disadvantages that made this technique an imperfect anti-spoofing method [66, 67].

3.3.5 RAIM/IMU

Inertial measurement unit (IMU) is an electronic device that contains a set of sensors such as accelerometers, gyroscopes, and sometimes magnetometers. IMU data is usually incorporated into an inertial navigation system (INS) to calculate the position, orientation, acceleration, and velocity of a moving object. The information obtained from IMU/INS sensors can be integrated into GPS to improve the accuracy of localization and to let GPS continue working in places where GPS signals are unavailable, such as tunnels and inside buildings. However, relying only on IMU sensors suffers from wrong initial GPS position information and accumulative errors [68]. Therefore, using the information of IMU could be more useful for localization purposes when it is merged with other methods like time-difference carrier phase, Bayesian estimation methods, and receiver autonomous integrity monitoring (RAIM) [69, 70, 71]. RAIM introduces a set of techniques that assess the integrity of GPS signals. Early RAIM techniques aimed to isolate a fault GPS signal by using redundant GPS pseudo-range measurements that are directly collected from all accessible satellites [72]. Newer RAIMs, however, incorporate the fault detection and exclusion (FDE) technique through combining RAIM with other methods [73, 74]. One popular method is to integrated RAIM with INS to provide additional redundant information from IMU sensors [75, 76]. However, the advantage of additional data may turn into a disadvantage if the sensors measurements are not true.

3.3.6 Vestigial Signal Detection (VSD)

The VSD anti-spoofing method is inspired by the difficulty of suppressing the authenticated GPS signal in the presence of spoofing signals. To eliminate the true GPS signal, a spoofer needs the centimeter accuracy knowledge of the 3D position phase difference between its antenna and the target's antenna, and also pico-second accuracy of its processing delay and the transmission delay. Without such accurate measurements by a spoofer, a vestige of

an authenticated signal will remain. A three steps procedure is proposed by authors [77, 78] to reveal a vestigial signal in the data (if it is present). In order to find the original signal, the VSD steps are repeated until a probability of detection threshold or some desired probability of false alarm are met.

It is worth noting that i) depending on the design, reliability, and the cost of a GPS component, one could select a technique to use, and ii) non of above techniques is complete. That is, each of them has some (tiny) drawbacks for detecting fake GPS signals. Hence, employing few techniques simultaneously could (nearly) complement them for each other. Knowing each technique, we will explain how to build a related graph and how to assign vulnerable probability for each node.

3.4 Modeling of Secure Autonomous Vehicles Using Bayesian Networks

In this section, we provide a theoretical model to measure the security of AVs. First, we describe a security model based on defense graphs for monitoring vulnerable components in an AV. Then, we explain how we consider threats and risk assessment for the model. Finally, we apply BN analysis as a simple but powerful tool to perform security measurements in this model.

3.4.1 Proposed Security Model

A security monitoring unit is an essential part of an AVs' central processor, which investigates all required data to assure the security of a vehicle. Figure 3.2(b) shows a typical security monitoring unit consisting of major attack surfaces [6]. Here, each attack surface is referred to as "component". As a part of processor, this unit has access to all required data for protection purposes.

In order to monitor the security status of an AV, we assess all vulnerable components. Let us define SV as the security state of an AV as follows:

$$SV \triangleq \{S_1, S_2, ..., S_n\}, \tag{3.3}$$

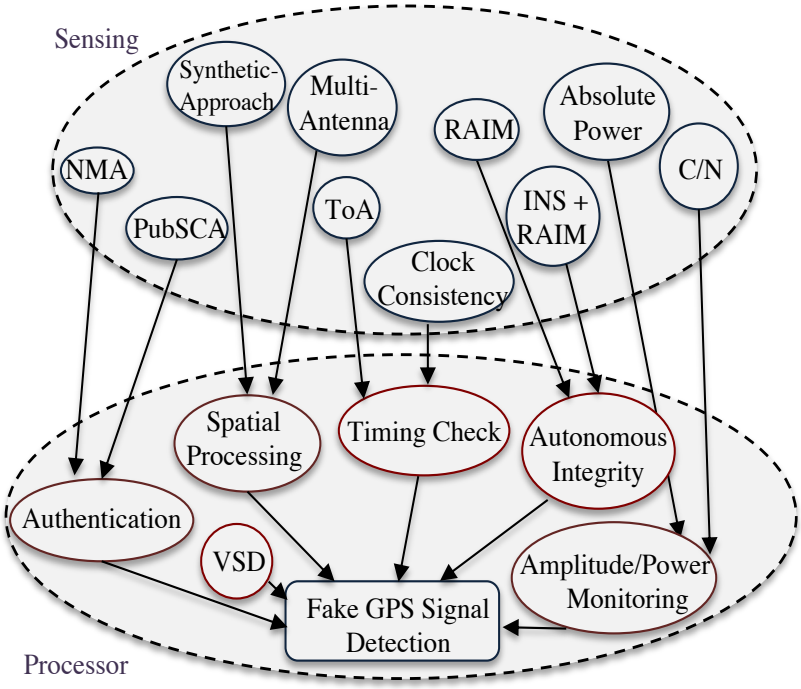


Figure 3.4: graphical model for a secure GPS component in AV.

where S_i denotes the security state of the ith vulnerable component. Each security state could be either normal or abnormal. A component is in an abnormal state when an attacker successfully mounts an attack on the component (i.e., the component is exploited). To ensure security, we could employ countermeasures for vulnerable components to prevent them from being exploited. Consideirng this point, we define a set of defense techniques as observable contexts to determine the security states as follows:

$$S_i = f(C_{i1}, C_{i2}, ..., C_{ik}). \tag{3.4}$$

Each observable context C_{ij} refers to the jth element of an defense technique related to the ith vulnerable component. To clarify this, consider Figure 3.4 as a graphical representation model for protecting a GPS component. Hence, this graph can be considered as a defense graph. As shown, we employ several techniques such as a timing check (v_i) in the processor to detect counterfeit GPS signals. Each technique needs some elements, such as clock consistency (w_1), to be accomplished. These predefined elements as part of defense techniques

provide observable contexts (C_{ij}'s). We utilize information from observable contexts and apply Bayesian inference as a mathematical reasoning method to characterize unobservable security states (S_i's). In the following section, we discuss threats against C_{ij}'s and the risk assessment of S_i's.

3.4.2 Threat Identification and Risk Assessment

Threat identification is the first step towards devising a security model for a system. In this work, we assume that vulnerable components of an AV have been already identified, thanks to previous works such as [6, 7, 8]. This allows us to employ defense graphs formed by countermeasures to protect AVs. Threats in the context of a defense graph could be interpreted as possible ways that counterfeit signals could go through the countermeasures without being detected. This means that a vulnerable component can be successfully exploited if none of corresponding countermeasures detect the fake signal. For instance, if an attacker remains undetected by the authentication countermeasure in Figure 3.4, it might be able to tamper with GPS information, causing a major threat. There exist several frameworks such as Microsoft's STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege) for threat identification that have been demonstrated to work well for AVs [79].

Once threats are identified, risk assessment could be carried out to determine the level of security in a system. There exist some methodologies to do the risk assessment, such as EVITA and CVSS (Common Vulnerability Scoring System). Risk assessment contains two fundamental parts: impact (or severity) of threats, and likelihood of threats. In order to estimate the impact of a threat, one could employ parameters that directly associate with harm to stakeholders. Safety, privacy of drivers, operational performance, and financial losses of a vehicle are four factors commonly used in automative risk models [80, 81]. The level of each factor can be categorized as none, low, medium, and high. To approximate the likelihood of a threat, one should calculate the probability of a successful attack. This could also be evaluated based on the above risk assessment models. For instance, expertise,

Table 3.1: Example of EVITA risk assessment factors: (a) Impact of an attack on GPS, (b) Likelihood of a threat against the ToA countermeasure.

(a)

	Safety	Financial	Privacy	Operational
GPS	High	Medium	High	Medium

(b)

	Expertise	Knowledge of Target	Window of Opportunity	Equipment
ToA	2	1	3	1

knowledge of target, window of opportunity (including time requirement), and equipment are four main parameters in EVITA to estimate the likelihood of threats. The level of each can be rated between 0 to 3. Table 3.1 shows examples of evaluation of impact and likelihood of threats.

Having the levels of impact and likelihood, we can compute the risk which is a function of both. A standard risk model can be defined as follows:

$$Risk = Likelihood \times Impact \tag{3.5}$$

where risk indicates risk for a set of countermeasures. The effect of countermeasures appears only in the likelihood, and not the impact, of a threat. Therefore, countermeasures directly affect the value of likelihood, while impact is determined by the functionality of a component (such GPS) within the AV. Also, the intrinsic uncertainty of attacks leads us to assess parameters based on probabilities. Here, *impact* can be directly estimated from the parameters in risk rating methodologies, such as Table I(a). To obtain *likelihood* for a component, however, two points should be considered: i) the quantity and the quality of the employed countermeasures and ii) cause-effect relationships between the elements of countermeasures. The former can be captured through standard parameters (e.g. Table I.(b)), and the later can be represented by directed acyclic graph (DAG) as a defense graph. Having the graph with related parameters enable us to infer likelihood using BN analysis.

3.4.3 Bayesian Network and Uncertainty

A Bayesian network is a graphical model for probabilistic inference that denotes the relationship between a set of variables by a directed acyclic graph (DAG). A BN is a pair (S, P), where S denotes a network structure, and P denotes a set of conditional probability distributions. Let us consider a DAG $S = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{v_1, v_2, ..., v_n\}$ represents a set of nodes, and $\mathbf{E} = \{e_1, e_2, ..., e_n\}$ represents a set of edges. Using this definition, each node could denote a countermeasure technique, such as time checking in Figure 3.4, or an element of it, such as clock consistency. A directed edge exists from node v_i to node v_j, only if there is the possibility for an exploit to be instantiated from v_i to v_j. Generally, in order to build a defense graph, the functionality of each node as well as cause-effect relationships between nodes (w.r.t. an application) must be captured in the BN framework.

Once we build a BN, we are able to perform probabilistic inference. Here, we are interested in applying marginal and posterior probability distributions to measure vulnerability for components. To clarify this, assume that we want to quantitatively measure the vulnerability of v_i that is shown in Figure 3.4. Assuming $\mathbf{W} = \{w_1, w_2\}$ as parent nodes of v_i, we can compute the following:

$$p(v_i) = \sum p(v_i|w_1, w_2)\ p(w_1, w_2), \tag{3.6}$$

$$p(v_i|\mathbf{W}) \propto p(\mathbf{W}|v_i)p(v_i). \tag{3.7}$$

Equation (3.6) is a marginal probability distribution to obtain prior probability for v_i, and equation (3.7) represents a posterior probability distribution using the prior probability and $p(\mathbf{W}|v_i)$ as a likelihood distribution. Using equations (3.6) and (3.7), we are able to calculate the likelihood of a successful attack on v_i, given the vulnerability of $\mathbf{W}$ (v_i's parent nodes).

Before applying the BN theory to obtain the security state of each vulnerable component, we need to capture uncertainties related to a realistic AV application. To this end,

Table 3.2: Prior probabilities of anti-spoofing techniques for detecting fake GPS signals using EVITA and CVSS.

	ToA	CLK-Cons.	NMA	PubSCA	C/N	Abs-power	RAIM	RAIM-INS	Multi-Ant	Syn-App	VSD
EVITA	0.58	0.50	0.75	0.83	0.58	0.75	0.42	0.75	0.66	0.58	0.83
CVSS	0.57	0.34	0.73	0.82	0.53	0.72	0.36	0.65	0.72	0.66	0.88

let us assume that Figure 3.3(a) shows a portion of a complete BN. Each node represents an anti-attack element to protect a vulnerable component. For instance, let us assume nodes A and B are two anti-attack elements for a secure component C. To yield a successful attack on node C, nodes A and B must have been unable to detect the attack. Hence, the framework for the reasoning of our defense graph is AND logic. Figure 3.3(b) indicates a conditional probability table (CPT) in which different scenarios of detection (D) and not detection (ND) are considered. The true (or false) state signifies a successful (or unsuccessful) detection on a component, respectively.

Here, we also account for the uncertainty between neighboring nodes due to their imperfect accuracy and trustworthiness. In addition, there exists an inherent uncertainty in attack structures. That is, even though an attack is successfully mounted on nodes A and B, there is no guarantee for the attacker to successfully carry out its attack on node C. To capture these points, we consider coefficients ζ_1 and ζ_2 between nodes, as shown in Figure 3.3(a). Considering these coefficients, we define θ_i in the CPT to indicate the probability of a true state in node C. In a defense graph, it is reasonable to have a high reliability between nodes, which implies small values for θ_i, $i = 1, 2, 3$, and a value close to 1 for θ_4.

In the next section, we investigate the security measurement of GPS signals as a vulnerable component. Other vulnerable components in an AV (e.g., LiDAR, camera) could be investigated in the same fashion.

Table 3.3: Example of conditional probability table

ToA	Clock Consistency	Timing Check	
		T	F
0.57 (D)	0.34 (D)	0.005	0.995
	0.66 (ND)	0.05	0.95
0.43 (ND)	0.34 (D)	0.10	0.90
	0.66 (ND)	0.995	0.005

3.5 Case Study: Secure GPS Component

GPS spoofing is among the highest threats for AVs. Hence, in this case study, we investigate the security measurement of GPS using the proposed BN model. In particular, we would like to obtain likelihood and risk for a defense graph shown in Figure 3.4.

3.5.1 Modeling and Parameterizing

A principle objective of this work is to quantify the security of a GPS component for AVs, by means of the following: (a) building a defense graph using BN model, and (b) parameterizing elements of the graph. Combining these two allows us to make an inference for likelihood, hence risk.

In order to model a defense graph for a GPS component, all possible ways to detect counterfeit GPS signals must be considered. Here, six most effective anti-spoofing techniques are selected. Each technique includes different elements for sensing abnormalities. Figure 3.4 shows a defense BN model for a GPS component obtained from cause-and-effect relationships among the elements of anti-spoofing techniques. These techniques are well studied in [82, 68, 63, 58, 77]. As can be seen, each technique (e.g., timing check) contains a few elements (e.g., clock consistency) to sense environment and send the required data for processing purposes. However, there is a possibility for an attacker to defeat an anti-spoofing technique which leads us to likelihood.

To find the value of likelihood, we need to determine the prior probability of each element and the conditional probability between the elements in the graph. We employ three approaches to make these evaluations: (a) EVITA as a risk assessment model, (b) CVSS that uses existing databases such as the National Vulnerability Database (NVD), and (c) several studies that have already addressed similar issues (e.g., [82, 68, 83, 84]). We apply the first two to find the prior probability and the last one to find the conditional probability. As we mentioned in section 3.4.2, we use four parameters for EVITA evaluation. For instance, as can be seen in Table I(b), since the summation of values is 7 and the total possible value is 12, we derive $\frac{7}{12}$ as the probability of detection for ToA. In CVSS, we consider two major concepts in calculating the scores: the base score (BS) and the temporal score (TS). The BS quantifies the intrinsic attribute of each vulnerability, which is independent of time and user environment. The TS, however, assesses the vulnerability based on properties that might change over time. Using BS and TS scores, the CVSS generates a value from 0 to 10 that can be simply converted to a probability by dividing the score over 10 [85]. Table 3.2 indicates the values of prior probabilities based on EVITA and CVSS. To obtain conditional probabilities between graph nodes, we use previous literature to consider all dependencies between anti-spoofing elements. We define four discrete probability levels w.r.t. the accuracy of anti-spoofing methods: 0.995 (almost sure), 0.99 (probable), 0.95 (highly expected), and 0.90 (expected). These values represent θ_is in the CPT table of Figure 3.3(b). For instance, Table 3.3 shows a CPT using CVSS for the timing check unit. CPTs for the rest of the anti-spoofing techniques can be obtained in the same fashion. Having a BN graphical model and its corresponding CPTs, the next step is to perform an inference to find likelihood for the GPS component.

3.5.2 Evaluation and Discussion

In what follows, we evaluate likelihood of threats and risks using equations (3.7) and (3.5). To obtain likelihood, we apply Bayesian inference. We initially determine the states of BN model and their roles for detection. It is shown in Figure 3.4 that there are 16 nodes,

Table 3.4: Likelihood of threats and risk probabilities for a sample of combinations of GPS anti-spoofing techniques.

Anti-attack GPS Techniques	CVSS		EVITA		Likelihood (EVITA) + Errors		
	Likelihood	Risk	Likelihood	Risk	1 %	5 %	10 %
Authentication (Aut)	0.0599	0.0499	0.0528	0.0440	0.0533	0.0554	0.0580
Aut, Timing Check (CT)	0.0362	0.0302	0.0250	0.0208	0.0300	0.0509	0.0823
Aut, CT, and Signal Processing (SP)	0.0098	0.0081	0.0071	0.0058	0.0087	0.0173	0.0334
Aut, CT, SP, and Amp/Pwr Monitoring (APM)	0.0022	0.0019	0.0014	0.0011	0.0018	0.0043	0.0104
Aut, CT, SP, APM, and RAIM/INS	0.0009	0.0008	0.0004	0.0003	0.0005	0.0015	0.0042
Aut, CT, SP, APM, RAIM/INS, and VSD	1.3×10^{-4}	1.1×10^{-4}	7.8×10^{-5}	6.5×10^{-5}	0.0001	0.0004	0.0013

each of which has two states that provide 2^{16} possible states. By employing CPTs such as Table 3.3, these states are reduced to 2^6. Then, we apply equation (3.7) to obtain the posterior probability of fake GPS signal detection (likelihood) given the incorporated anti-spoofing techniques. Assuming $impact = 0.833$ given by Table 3.1(a) for a GPS component, we can derive the risk defined in (3.5).

Table 3.4 shows resulted beliefs for likelihood and *risk*. Since all the 2^6 states could not be shown here, a few combinations are selected. It can be seen that the likelihood and the risk of threats are generally decreased by utilizing a higher number of countermeasures. For instance, based on EVITA, the likelihood of attack could be reduced from 5.3% to less than 0.1% and 0.01% by using, respectively, five and six anti-spoofing techniques instead of just one. As can be noted, results for CVSS and EVITA are close to each other. This is not surprising, as the prior probabilities of anti-spoofing elements (Table 3.2) are also close. This type of analysis could also help in choosing the number and type of anti-attack techniques to be deployed in the presence of energy, size, and cost limitations. Furthermore, in order to study the resilience of the proposed model, the likelihood of threats is evaluated for different levels of errors. The cause of these errors could vary from noise and inaccurate processing of data to hardware problems in deployed countermeasures. It can be seen that

threat likelihood, hence the risk, can be contained to small values, particularly for small errors, when five or more countermeasures are present.

3.6 Summary

We have introduced a framework using a Bayesian defense graph to study the cybersecurity of AVs. In particular, we have employed risk assessment models such as EVITA to study the threat likelihood and risk for vulnerable components in AVs in the presence of countermeasures. In a case study, we have applied this framework to infer a belief for the likelihood of threats and risks for GPS signals. Our results confirm that the likelihood of threats can be reduced to 0.01% depending on what anti-spoofing techniques are employed. Future work will focus on the impact of cooperation between vehicles to improve the security of an AV.

CHAPTER 4

MISBEHAVIOR DETECTION IN EPHEMERAL NETWORKS: A LOCAL VOTING GAME IN PRESENCE OF UNCERTAINTY

Emerging short-lived (ephemeral) connections between wireless mobile devices have raised concerns over the security of ephemeral networks. An important security challenge in these networks is to identify misbehaving nodes, especially in places where a centrally managed station is absent. To tackle this problem, a local voting-based scheme (game) in which neighboring nodes quickly decide whether to discredit an accused (target) node in mobile networks has been introduced in the literature. However, nodes' beliefs and reactions significantly affect the outcome of target node identification in the collaboration. In this work, a plain Bayesian game between a benign node and a target node in one stage of a local voting-based scheme is proposed in order to capture uncertainties of nodes for target node identification. In this context, the expected utilities (payoffs) of players in the game are defined according to uncertainties of nodes regarding their monitoring systems, the type of target node and participants, and the outcome of the cooperation. Meanwhile, incentives are offered in payoffs in order to promote cooperation in the network. To discourage nodes from abusing incentives, a variable-benefit approach that rewards each player according to the value of their contribution to the game is introduced. Then, possible equilibrium points between a benign node and a malicious node are derived using a pure-strategy Bayesian Nash equilibrium (BNE) and a mixed-strategy BNE, ensuring that no node is able to improve its payoffs by changing its strategy. Finally, the behavior of malicious and benign nodes is studied via simulations. Specifically, it is shown how the aforementioned uncertainties and the designed incentives impact the strategies of the players and, consequently, the correct target-node identification.

The remainder of this chapter is organized as follows. Section 4.1 provides a short background for game theory. The introduction of the work is provided in presented in section

4.2. Section 4.3 devotes for a brief overview of the related work and elaborates on what distinguishes our work from the rest. Section 4.4 describes the proposed system model that includes a game-theoretic model, payoff design, and an adaptive benefit scheme in the game. Section 4.5 applies Bayesian game analysis to derive equilibrium points in the proposed model. Section 4.6 is devoted to extensive numerical results. Section 4.7 summarizes the chapter.

4.1 Background

Game theory is the branch of mathematics concerned with the analysis of strategies for dealing with competitive situations where the outcome of a participant's choice of action is critically dependent on the actions of other participants. Game theory has been applied in different contexts from computer science to business and biology [86]. It provides accounting tools to ease the study of interdependent situations where the logic becomes very complicated.

To begin, a famous simple game called prisoner dilemma is presented. In this game, it is presumed that two suspects are arrested. The police think that the subjects were trying to rob a store. However, they can only prove that the suspects were trespassing. Hence, the police need one of the suspects to rat out the other. Here, three scenarios could happen while they are in custody. First, none of the suspects confesses to the robbery. In this case, the punishment will be one month of jail time for each, simply due to trespassing. Second, one suspect confesses and one does not. In this case, the police reward the one who confesses with his/her freedom but will punish the suspect who kept quiet with 12 months of jail time. Finally, if both of suspects confess, then each of them will get 8 months of jail time. These numbers are referred to as payoffs of the game. All scenarios are shown in Figure 4.1. It is obvious that each thief wants to minimize his/her own punishment. Now, the question becomes what will happen? To check this, assume that player 1 (P1) keeps quiet. Knowing this, player 2 (P2) would confess rather than remaining silent, because 0 month of jail time is better than one month of jail time for him. Consequently, P1 receives 12 months of jail

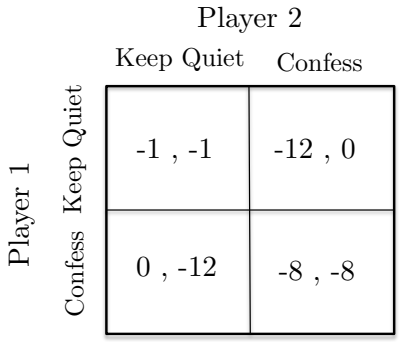


Figure 4.1: Prison dilemma game.

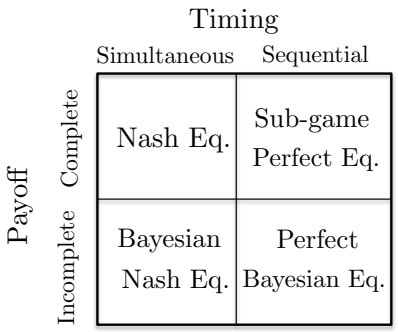


Figure 4.2: General types of games.

time, while P2 is freed. This scenario is shown in the top right corner of Figure 4.1. Based on this fact, P1 decides to confess. In this scenario, P2 confesses instead of remaining silent, because 8 months of jail time is better than 12 months of jail time. Therefore, both players confessing means that each of them gets 8 months of jail time. This is shown in bottom right corner of Figure 4.1. This strategy is referred to as Nash equilibrium (NE). NE is actually a set of strategies, one for each player, such that no player has incentive to change his/her individual strategy. NE is inherently stable and only cares about individual deviations rather than group deviations. It is worth mentioning that if the summation of all payoffs for each player in a game is zero, then the game is called a zero-sum game.

Games can be divided into four general groups based on information and timing. The information of a game is complete, if the payoffs are known for all players, otherwise information is incomplete. Players can also play simultaneously or sequentially. Figure 4.2 shows all types with solution strategies for each case. The prison dilemma explained above is an example of complete simultaneous game which led to a NE solution for both players. In what follows, we describe a sequential game that is based on a local voting-based game. This game has been used to recognize and revoke malicious nodes in ephemeral vehicular networks [11, 12, 13].

4.2 Introduction

4.2.1 Motivation

The proliferation of temporal peer-to-peer communication between wireless devices gives rise to the formation of short-lived (ephemeral) networks due to the unpredictable existence of mobile nodes. Ephemeral networks are pervasive over a variety of applications, such as vehicular ad hoc networks, mobile social networks, wireless sensor networks, etc. [87, 88, 89, 90]. These networks are attractive to malicious nodes so they join and manipulate sensed data, thereby influencing network performance [91]. For example, in the case of vehicular ad hoc networks (VANETs), a malicious vehicle can inject false information to its neighbors and trigger a serious problem on the road. In addition to data manipulation or sending false information, a malicious node can compromise the vehicle's routing efficiency by not forwarding the packets that it received in the network. Therefore, an important security improvement in ephemeral networks is to reveal the type of nodes, either benign or malicious, especially in places where centrally managed stations are absent. In such transitory distributed networks, quick cooperation among neighboring nodes can provide effective solutions toward improving network performance [92, 93]. However, nodes are usually selfish and reluctant to cooperate, simply because they must use their resources. In addition, each node has some inherent uncertainties in a collaboration, including the type of participants, the accuracy of its own components (e.g., detection system), the value of its contribution, and attainable outcomes, all of which affect the node's decision about whether to participate. In this respect, it is crucial to provide incentives according to different reactions of nodes under uncertainty in order to achieve malicious-node detection.

4.2.2 Related Work

A variety of precise and effective schemes have been proposed to detect misbehaving nodes. The authors in [94] [95] have provided a comprehensive literature review on misbehavior detection in cyber-physical systems (CPSs) and intelligent transportation systems, which covered a variety of solutions for transient networks wherein nodes have different limitations.

In particular, Liu et al. [96] studied the interactions between an attacker and a defender in order to detect misbehaving nodes in ad hoc networks. They considered scenarios where a malicious node can either attack or not attack a benign node, while the defender may be in a monitoring or non-monitoring state. The interactions between players were analyzed by game theory, which is a powerful tool to obtain the best strategies of independent decision makers [97]. Although this work clearly described the individual interactions and uncertainties between two agents, it did not consider the identification of a misbehaving node for all nodes in the network. Another approach for misbehavior detection is to use reputation systems for which a history of credits is built for nodes based on their past behavior in the network. However, this method needs a database that can be created over time, and nodes may have to continuously monitor their neighbors [98]. Considering short time connections between nodes, reputation systems do not appear to be a proper approach in ephemeral networks [11].

In other work, a local revocation process is introduced to consider the dynamic nature of ephemeral networks [11, 13, 99, 100, 101, 102, 103, 104]. In the revocation process, a benign node as an initiator is assumed to detect (or become suspicious of) a malicious node and broadcasts its identification (ID) as a *target node* or an accused node. Then, other benign neighboring nodes run a local voting-based scheme to decide whether to discredit the target node. Scholars in [11, 13, 99] studied the local revocation process as a sequential voting game in which a benign node can choose one of three strategies with regard to (w.r.t.) a target node: voting, abstaining, or self-sacrificing. A benign node chooses between a voting strategy or an abstaining strategy based on its economic considerations in the game. Also, the benign node might use a self-sacrificing strategy to declare the invalidity of its current identity as well as the identity of the target node. Some limitations of the proposed revocation process in vehicular ad hoc networks (VANETs) have been pointed out by Liu et al. [99]. For example, they underlined the assumption of complete information for nodes in the game and

proved that the identification of the target node may not be possible without considering the rate of false positives and the rate of false negatives.

To address existing problems and introducing new approaches for misbehavior detection in mobile networks, Abass et al. [100] introduced an evolutionary game wherein all benign nodes cooperate in the voting game, focusing on unsuccessful revocation and over-reacted revocation decisions. Scholars [102, 101] have developed a weighted voting game based on clustering architecture to effectively solve the problem of false accusation. Masdari [103] proposed a collaborative false accusation approach to stop wrong accusations in the network. Diakonikolas and Pavlou [105] emphasized the inverse power index problem in designing weighted voting games and proved that the problem is computationally intractable for a broad family of semi-values. In another work, Subba et al. [106] proposed an intrusion detection system (IDS) that employs the concept of election leaders and a hybrid IDS, with the aim of avoiding the continuous monitoring of nodes in mobile ad hoc networks (MANETs). These authors then extended their work in [107] by providing a multi-layer game theory to address the problem of dynamic network topologies in VANETs. While these efforts are effective at minimizing the volume of IDS traffic, they do not address the uncertainty of nodes alongside incentives in local voting games. Others [108] have studied the impact of incentives on misbehavior detection in unmanned aerial vehicle (UAV)-assisted VANETs. Silva et al. [109] proposed a voting scheme to generate a large set of novel strategies from available expert-based ones. The proposed scheme selects the best strategies while the model of opponents are considered during the game. However, this work does not focus on identifying a malicious node in an ephemeral network.

4.2.3 Novelties and Contributions

Some differences distinguish this work from other significant contributions in the literature. First, we design a local voting game wherein the type of target node can be either malicious or benign. This is in contrary to several papers [13, 100, 103], in which authors

presume that a benign initiator broadcasts the ID of a malicious node as the target node. The reason of our assumption is that every node, including malicious nodes, can accuse others and all benign nodes do not necessarily need to know each other in the network. In our design, nodes vote regarding the type of target node based on the accuracy and cost of their monitoring systems as well as a pre-defined belief about the portion of malicious nodes in the network. These considerations provide a framework in which nodes look at their resources and potential hostile environments before taking part in the voting game. Second, we consider that benign nodes are uncertain about the strategy of malicious nodes in the network. This is because a malicious node might intentionally not attack a benign node in order to obtain its support during the local voting game. This point is omitted in the design of the local voting game in [11, 13, 99, 100, 101, 102, 103]. In particular, these works mainly focused on target node identification based on the rate of participation rather than considering the malicious node strategy of avoiding being accused. Third, we design incentives to encourage only knowledgeable nodes (i.e., nodes that have already monitored the target node) in the game. Otherwise, the incentives lead to many random votes in the game, which might spoil the result of cooperation. This point is mostly overlooked in the above literature, and [110] that studied the role of nodes' incentives in order to contribute to ephemeral social networks. Fourth, we consider that both benign nodes and malicious nodes can take part in the voting game. This implies that a benign node cannot rely solely on others' votes, owing to misleading votes from malicious nodes. The incomplete information of nodes in voting was not studied in the above literature, including [11, 100, 103]. Fifth, the cost of a group participating in the game (a.k.a., social cost) should be designed based on nodes' contributions and their uncertainties about the results. For example, a cooperative node should be punished less than an abstaining node when the collaboration becomes unsuccessful. Finally, a potent design should provide more rewards for decisive votes. Although some authors studied weighted voting games [102], [111], they mainly based

their designs on clustering heads rather than the value of a vote in the middle of a local voting games.

We implement the points above in analyzing misbehavior detection using the local voting-based scheme in the presence of uncertainty. Our main contributions in this work can be summarized as follows:

- We provide a game layout between a benign player and a target node in one stage of a local voting game using a static Bayesian game in order to detect misbehaving nodes in an ephemeral network. In this regard, we design expected utilities (payoffs) that capture uncertainties (explained above) regarding detection systems, types of participants, type of target node, and outcome of the game. We offer incentives in node payoffs in order to promote cooperation in the network. Furthermore, we introduce a variable benefit for cooperative nodes in which rewards are adjusted according to the value of contributions in the game. This scheme prevents nodes from abusing incentives by pointless participation.

- We derive possible equilibrium points between a benign player and a target node in the game using a pure-strategy Bayesian Nash equilibrium (BNE) and a mixed-strategy BNE, ensuring that no node can improve its utility by changing its strategy. The best strategies can be adopted by malicious nodes and benign nodes w.r.t. different game parameters.

- We provide extensive numerical results to verify the analysis and investigate the impact of cooperation parameters and uncertainties on the identification of malicious nodes. Our results confirm the influence of the designed incentives, hence participation rate, on the strategies of malicious and benign nodes in an ephemeral network. We observe, in particular, that if the participation incentives go beyond a certain limit, then *correct* target-node identification will be decreased, in spite of the growing participation rate.

4.3 Assumptions and Problem Description

4.3.1 Network Model

We study misbehavior detection in a network where nodes have short-lived connections, and a centrally managed station is absent. We use a vehicular ad hoc network (VANET) as a typical example of an ephemeral network to explain our approach. We assume that nodes (e.g., vehicles) are powerful enough to have wireless communication among themselves. We consider a contention-based medium, e.g., IEEE 802.11p in a VANET, which can represent the nature of wireless channel access [11]. We further assume that a base station or a certificate authority has already established the credential of the nodes; hence, each node has a unique ID.

We assume that there are two types of nodes in the network: malicious and benign. Malicious nodes may attack benign nodes by disseminating false information. For example, a malicious car might inject faulty data to the sensors of the car that follows it, in order to manipulate an optimal space between them [112]. On the other hand, a benign node is equipped with a monitoring system to detect abnormal or counterfeit signals. For example, an autonomous vehicle can use a set of anti-spoofing techniques to detect fake global positioning system (GPS) signals [17]. However, benign nodes do not necessarily need to monitor all of their neighbors, due to the cost of monitoring over all short-lived connections.

4.3.2 Local Voting Game

Nodes can participate in a local voting-based scheme (game) in order to determine the identity of a node in the network. The voting game starts when an initiator broadcasts the ID of a target node. Then, neighboring nodes choose either to vote or not to vote (abstain) on type of the target node. Each node calculates its costs and benefits in order to choose a strategy. Nodes can broadcast their decisions sequentially, and each node's decision is made in one stage of the game. We assume that the belief of a node w.r.t. the target node is independently inferred and does not change (e.g., by other votes) during the game. This is because nodes are uncertain about the correctness of other votes. In other words, a node may

not know the types of all previous nodes that have already voted. Also, the node is unaware of the target node's strategy w.r.t. previous nodes. Hence, we assume that a node votes based on its own detection system. It is worth noting that monitoring all neighboring nodes and their interactions might be too costly for a node in an ephemeral network. The target node is identified when the number of votes in one type (either malicious or benign) reaches a pre-defined number. This number is denoted by n_{th} and is studied in section 4.6.2 . If correct (wrong) votes reach n_{th}, then we will have correct (wrong) target node identification. If n_{th} is not reached during the game, then we will have *undecided* target node identification.

Malicious nodes and benign nodes can choose some strategies in the game. A malicious node can select to attack or not to attack a benign node, while it is unaware of being monitored. After a target node is determined, a benign node checks whether it has already monitored the target node. If it has not monitored the target node, it will abstain from voting, simply because it does not have any information about the node. But, if the benign node has monitored the target node, it will calculate its payoffs. If its voting payoff outweighs its abstaining payoff, then the benign node will vote; otherwise, it will abstain. On the other hand, malicious nodes vote against a benign target node and for a malicious target node. If there is no possibility for malicious nodes to change the result of the game in their favor, then they abstain from voting. We do not consider strategic malicious nodes that can optimize their types of votes to collect some credits or to send multiple wrong votes (e.g., Sybil attack [113]).

4.3.3 Problem Definition

Figure 4.3 shows an example of a local voting game in a VANET that helps us explain the problem. As can be seen, nodes $0, 2$, and 5 are malicious, and the other nodes are benign. We assume that node 0 is the target node and $n_{th} = 2$. We also assume that node 1 casts a correct vote, and node 2 casts a wrong vote. Now, node 3 should reveal its strategy. Here, we study the possible reactions of node 3 (as an example of a benign node) in the game where identification of the target node is not yet finalized. If node 3 has not monitored node 0,

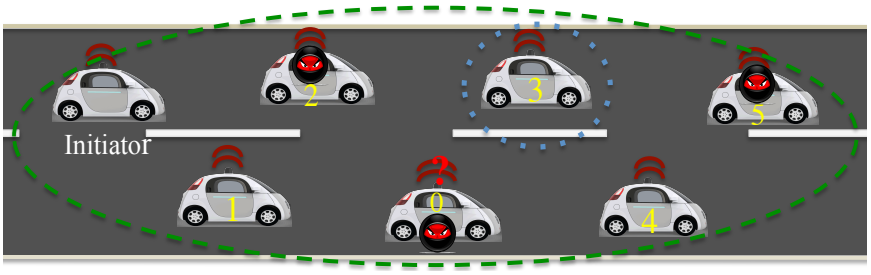


Figure 4.3: Example of local voting game in VANETs.

then it simply abstains. Otherwise, if it has already monitored node 0, it can select between voting or abstaining. To choose its strategy, node 3 faces some uncertainties that greatly impact its decision.

The first uncertainty of node 3 is about its own monitoring system, which has a predefined detection rate and false alarm rate. For example, the monitoring system of node 3 might recognize node 0 as benign, even though it is malicious. Next, node 0 might choose not to attack node 3 in order to produce a wrong perception. Hence, node 3, even if not attacked by node 0, would still be uncertain about the type of node 0. In addition, node 3 might not have monitored nodes 1 and 2; hence, it cannot vote based on previous votes. Moreover, node 3 may not have monitored all remaining nodes in the voting game, so it cannot count on their correct votes. This implies that the node is also uncertain about the outcome of the game, which potentially affects its decision. Nevertheless, incentives should be offered to encourage node 3 (and others) in cooperation. This could also avoid the formation of free-riders (i.e., nodes that benefit from the results without making any contributions [114]) in the network. It is worth noting that, depending on the stage of the game, a node's strategy could make a different level of impact on the results. For instance, if node 3 abstains, then the vote of node 4 is absolutely necessary for correct target identification ($n_{th} = 2$). Hence, incentives should be offered w.r.t. the value of a contribution.

On the other hand, malicious nodes are aware of an existing voting game in the network. That is, a malicious node knows that it might become a target node. The objective of a malicious node is to maximize the level of its aggressiveness in the network without being

identified. However, it is uncertain about being monitored by a benign node, and the strategy of a benign node in the game (i.e., voting or abstaining). In contrast, a benign node knows that some of its neighbors may be malicious. The objective of a monitoring benign node is to choose a strategy with the aim of target node identification. However, a benign node has some limitations in its detection system. Also, it is uncertain about the strategies of malicious nodes and, therefore, is uncertain about the type of the target node. Taking these points into consideration, our goal is the following:

- To design payoffs for a benign node w.r.t. the explained uncertainties and the value of its contribution in the game,
- To determine the best strategies for malicious nodes and benign nodes.

We address the first problem in section IV by considering the following: (i) the vote of a benign node that could be either correct or incorrect; (ii) the probability of correct target node identification in each stage, which is mainly based on votes that have already been cast; and (iii) the impact of a benign node's strategy on correct, wrong, and undecided target node identification. We address the second problem in section V. In particular, we develop one stage of the voting game as a Bayesian game to study the reactions of a benign node w.r.t. a target node. This helps us understand the best strategies of both types of nodes in the network.

To find the best strategies of players in the game, one could employ the concept of subgame perfect equilibrium, wherein nodes have complete information in a sequential game. However, as explained above, complete information does not seem to be a realistic assumption for nodes in an ephemeral network. Hence, we do not apply this concept to our model. Another solution is to use the concept of perfect Bayesian equilibrium (PBE) in dynamic Bayesian games to capture the incomplete information of nodes. In this context, a node needs to update its belief regarding its opponent's type according to the game evolution. However, we believe that the belief of a benign player regarding the type of target node should

not change by other votes during the game because of existing uncertainties about the types of neighboring nodes, their interactions, and the cost of monitoring over all neighbors in an ephemeral network. In principle, we assume that nodes are relying on their own detection systems to vote on the type of the target node. In this regard, we can not properly apply the concept of PBE to our model.

On the other hand, a Bayesian game is general enough to capture many scenarios between attackers and defenders in one stage of the game [114]. In particular, we can consider a potential attacker as a target node that chooses to attack or not to attack a benign node, and the benign player that can vote or abstain in one stage of the game. One advantage of applying Bayesian games to our model is that nodes independently infer the type of a target node using a detection system, and they do not need to know the type of all neighbors and their interactions with each other. In other words, a node can efficiently select its strategy according to the BNE that maximizes its expected utility with a set of parameters. However, one drawback of using BNE is that a benign node requires a sensible prior belief regarding the type of neighboring nodes. We let μ denote the prior belief of benign nodes about the portion of malicious nodes in a network (parameters are described in section 4.4.1). In practice, a node should be able to adjust the values of μ according to its knowledge of an environment. Nodes could assign a high value for μ in potentially hostile places such as crowded areas in mega cities, and a low value for μ in safe places such as rural areas.

4.4 Problem Formulation

In this section, we first introduce a set of parameters that are required to define the underlying game. Next, we provide some definitions to facilitate the explanation of our model. Then, we design payoffs that include individual beliefs and available voting information in the game. Finally, we propose a variable-benefit scheme that rewards participants according to the value of their contributions.

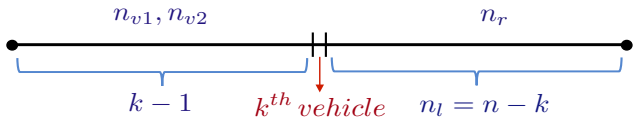


Figure 4.4: k^{th} stage of the game, where n_{v1}, n_{v2}, and n_r denote the number of correct, incorrect, and remaining votes, respectively, and n_l refers to the total number of nodes left to vote.

4.4.1 Parameters

To describe our model, we need to define a set of given parameters to design player's payoffs in the game. We list these parameters in Table 4.1. To begin, we assume that a benign node holds an asset with a security value of w, where $w > 0$. This parameter is considered in order to count for the value of a vulnerable asset in benign nodes that can be exploited by malicious nodes. A malicious node could compromise the asset by paying the cost of an attack, denoted by c_a. In contrast, a benign node protects its asset by monitoring for attacks, with probability P_m. This monitoring costs c_m for the node, and all costs are positive. It is rational to assume that $w > c_a$ and $w > c_m$. Otherwise, the attacker and the benign node lose their motivation to attack and to protect the asset, respectively. A benign node assigns a prior probability of μ for its neighbors to be malicious. We measure the performance of the monitoring system by considering the following: (i) α, which represents the detection rate (i.e., true positive rate), and (ii) β, which denotes the false alarm (i.e., false positive rate) for detecting abnormalities. It is sensible to expect that $\alpha > 0.5 > \beta$.

It is assumed that n nodes are in a neighboring area. Each benign node can vote by paying c_v as the cost of voting. The benefit of a correct strategy and the punishment of an incorrect strategy for a benign node are denoted by b and $-b$, respectively. To generalize the analysis, we design the game at one stage, say the k^{th} stage, in which the type of a target node has not yet been determined. Until this stage, it can be assumed that n_{v1} votes in one type (e.g. malicious) and n_{v2} votes in another type (e.g. benign) have already been cast for the type of target node. We let n_r denote the number of remaining votes required to identify the target node. Therefore, $n_r = n_{th} - n_{v1}$ or $n_r = n_{th} - n_{v2}$, depending on the belief of the

Table 4.1: List of parameters in alphabetical order.

Symbols	Meaning
α	Probability of detection (true positive)
β	Probability of false alarm (false positive)
μ	Prior probability of node being malicious
λ	Probability of a remaining node stays in the neighborhood
b	Benefit of correct strategy
$-b$	Punishment of incorrect strategy
c_a	Cost of attack
c_{gb}	Cost of group for incorrect identification of benign target node
c_{gm}	Cost of group for incorrect identification of malicious target node
c_m	Cost of monitoring of an asset
c_v	Cost of voting
n	Total number of nodes
n_l	Number of nodes left at k^{th} stage
n_r	Number of required votes at k^{th} stage to identify target node
n_{th}	Number of required votes to identify target node
n_{v1}	Number of correct votes for target node
n_{v2}	Number of incorrect votes for target node
p_k	Probability of successful target identification at k^{th} stage
P_m	Probability of monitoring
q	Probability of attack for malicious PLT
s	Probability of voting for monitoring PLB
w	Value of an asset

k^{th} node on the type of target node. Also, there are n_l nodes left in the game. Figure 4.4 helps us understand these parameters. We use p_k to denote the probability of correct target node identification at the k^{th} stage. It is assumed that the cost of the group (neighboring nodes) for the incorrect identification of a malicious target node and a benign target node are c_{gm} and c_{gb}, respectively.

A malicious node can choose to either attack or not attack a benign node based on its information in the game. In this regard, the probability of attack is considered and it is denoted by q in the analysis. Likewise, a node can choose to either vote or not vote. This is captured by defining the probability of voting and is denoted by s in the analysis. It is worth noting that a node considers different parameters to select its strategy, including

the accuracy of the monitoring system, costs, benefits, punishments, and the probabilities introduced here. Equipped with these parameters, we are able to design the expected payoffs for players in the game.

4.4.2 Game Definition and Notations

Here, we provide some definitions in order to facilitate the description of our model.

Definition 1: A Bayesian game G is defined by a tuple (N, A, Θ, μ, U), where N is a set of players, A is a set of actions, Θ is a set of players' types, μ is a common prior, and U is a set of utilities. These are defined as follows:

- N = (PLT, PLB), where PLT is a target node and PLB is a benign player.
- $A = (a_{PLT}, a_{PLB})$, where $a_{PLT} \in \{$attack, not attack$\}$, and $a_{PLB} \in \{$vote, abstain$\}$.
- $\Theta = (\theta_{PLT}, \theta_{PLB})$, where $\theta_{PLT} \in \{$malicious, benign$\}$, and $\theta_{PLB} \in \{$benign$\}$.
- Prior belief: PLB assigns μ for PLT being malicious, and PLB's type is common knowledge.
- $U = (u, v)$, where u refers to PLB's payoff, and v refers to PLT's payoff.

Figure 4.5 shows the strategic form of the game G, where rows and columns indicate the actions of a target node and a benign player, respectively. As can be seen, each window includes pairs of payoffs (u_z, v_z), where $1 \leq z \leq 9$, and the subscript z refers to the actions of both PLB and PLT in one scenario. For example, (u_1, v_1) at top left window in Figure 4.5 corresponds to voting PLB and attacking PLT. As can be seen in Figure 4.5, we need to define u_z and v_z to provide the layout of the game.

Definition 2: Each player's payoff can be defined as the summation of an individual payoff and a group payoff as follows:

$$u_z = u_{z,i} + u_{z,g}, \tag{4.1}$$

$$v_z = v_{z,i} + v_{z,g}, \tag{4.2}$$

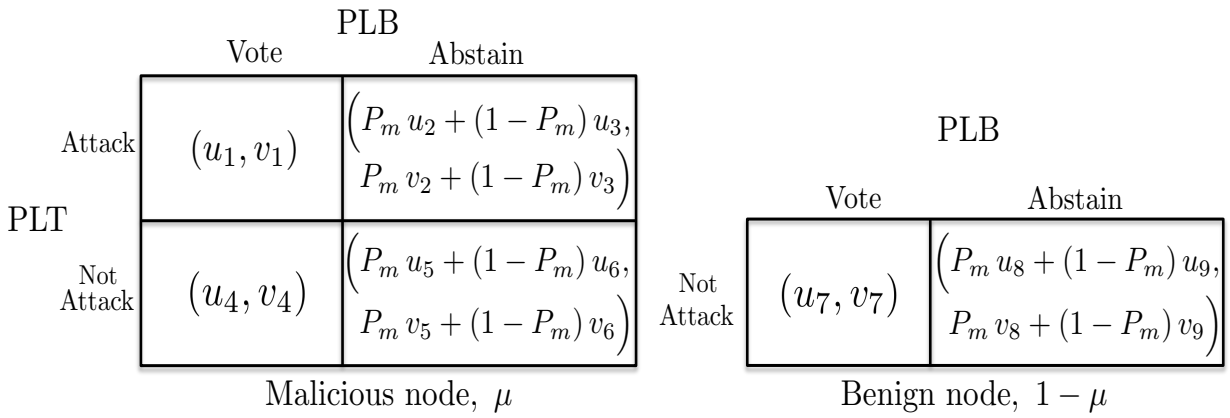


Figure 4.5: Players' payoffs in the game relative to a benign player (PLB) and malicious or benign target node (PLT).

where $u_{z,i}$ and $v_{z,i}$ denote individual payoffs, and $u_{z,g}$ and $v_{z,g}$ denote group payoffs. The individual payoff only considers interactions between two players, while the group payoff accounts for the impact of a player's strategy on all members in the neighborhood.

It is worth noting that the group payoffs capture the impact of a node's strategy on the security of all nodes in the network. This comes from a node by either voting or abstaining in the voting game. Group payoff also provides a framework to capture incentives for promoting cooperation in the game. In particular, the collaboration of nodes for malicious node identification would reward them, while abstaining from the collaboration could punish them in the game. These rewards and penalties are included in the group payoff of a node, because they relate to actions that impact all nodes in the network. The following section describes the design of both individual and group payoffs in detail.

4.4.3 Payoff Design

In what follows, we describe individual payoffs and group payoffs and find all u_z and v_z shown in Figure 4.5. Individual payoffs account for the costs and benefit of monitoring and non-monitoring, whereas group payoffs account for the costs and benefits of voting or abstaining from the game. The incentives of the game, including benefits and punishments, are considered in the group payoffs. This means that a node measures its group payoff before choosing its strategy. Initially, we focus on obtaining individual payoffs, i.e., $u_{z,i}$. Then, we

explain group payoffs, i.e., $u_{z,g}$. Finally, we use definition 2 to add individual and group payoffs to obtain all u_z and v_z.

Individual payoffs

To study individual interactions between two nodes, note that a malicious node could choose either to attack or not to attack a benign node. Also, the benign node could be either in a monitoring state or a non-monitoring state. Hence, we have four possible scenarios between a malicious node and a benign node:

- A monitoring PLB faces an attacking PLT.
- A non-monitoring PLB faces an attacking PLT.
- A monitoring PLB faces a non-attacking PLT.
- A non-monitoring PLB faces a non-attacking PLT.

What follows is a study of the payoffs for these scenarios to evaluate related $u_{z,i}$ and $v_{z,i}$.

Scenario I: A monitoring PLB faces an attacking PLT. Here, the monitoring PLB pays $-c_m$ as the cost of monitoring. However, it gains $(2\alpha - 1)w$ from its detection system. This is because the expected gain of the PLB relates to the true positive rate (α) and the false negative rate $(1 - \alpha)$ of its detection system, i.e., $\alpha w - (1 - \alpha)w$. Therefore, the individual payoff of a monitoring PLB in this scenario is equal to $-c_m + (2\alpha - 1)w$. This payoff corresponds to $u_{1,i}$ and $u_{2,i}$, where the PLT attacks a monitoring PLB. Hence,

$$u_{1,i} = u_{2,i} = -c_m + (2\alpha - 1)w. \tag{4.3}$$

On the other hand, the PLT pays $-c_a$ as the cost of the attack. The loss of the PLT can be assumed as the negative gain of the PLB's individual payoff [96], i.e. $-(2\alpha - 1)w$. Therefore, the individual payoff for an attacking malicious PLT in this scenario equals $-c_a -$

$(2\alpha - 1)w$. This payoff corresponds to $v_{1,i}$ and $v_{2,i}$. Hence,

$$v_{1,i} = v_{2,i} = -c_a - (2\alpha - 1)w. \tag{4.4}$$

Scenario II: A non-monitoring PLB faces an attacking PLT. Here, the non-monitoring PLB does not pay the cost of monitoring but loses its asset as a result of being in a non-monitoring state and attacking the PLT. Therefore, the individual payoff of a non-monitoring PLB in this scenario is equal to $-w$. This payoff corresponds to $u_{3,i}$. Hence,

$$u_{3,i} = -w \tag{4.5}$$

On the other hand, the PLT pays $-c_a$ as the cost of the attack, and its gain is equal to the loss of the non-monitoring PLB, i.e. $+w$. Therefore, the individual payoff for an attacking malicious PLT in this scenario equals $-c_a + w$. This payoff corresponds to $v_{3,i}$. Hence,

$$v_{3,i} = -c_a + w. \tag{4.6}$$

Scenario III: A monitoring PLB faces a non-attacking PLT. Here, a monitoring PLB pays $-c_m$ as the cost of monitoring. However, since the PLT is in a non-attacking state, any possible detection comes from its false alarm rate, i.e., $-\beta w$. Therefore, the individual payoff of a monitoring PLB in this scenario is equal to $-c_m - \beta w$. This payoff corresponds to $u_{4,i}$, $u_{5,i}$, $u_{7,i}$, and $u_{8,i}$. Hence,

$$u_{4,i} = u_{5,i} = u_{7,i} = u_{8,i} = -c_m - \beta w. \tag{4.7}$$

The non-attacking PLT does not gain or lose in the interaction. Thus,

$$v_{4,i} = v_{5,i} = v_{7,i} = v_{8,i} = 0. \tag{4.8}$$

Scenario IV: A non-monitoring PLB faces a non-attacking PLT. Here, since there has been neither an attack from the PLT nor monitoring from the PLB, we have 0 payoff. This payoff corresponds to $u_{6,i}$, $u_{9,i}$, $v_{6,i}$, and $v_{9,i}$. Hence,

$$u_{6,i} = v_{6,i} = u_{9,i} = v_{9,i} = 0. \tag{4.9}$$

Having $u_{z,i}$ and $v_{z,i}$ in hand, we continue defining $u_{z,g}$ and $v_{z,g}$ to be able to use definition 2 and obtain all payoffs.

Group payoffs

To study group payoffs, note that a monitoring PLB can choose between voting and abstaining on the type of PLT in the game. A non-monitoring PLB always chooses to abstain from voting because it does not have any information about PLT. In this respect, we can study group payoffs by considering four scenarios:

- Monitoring PLB faces an attacking malicious PLT.
- Monitoring PLB faces a non-attacking malicious PLT.
- Monitoring PLB faces a (non-attacking) benign PLT.
- PLB is in a non-monitoring state.

In the first three scenarios, we study the voting and abstaining strategies of a monitoring PLB w.r.t. the probability of correct target node identification (p_k) in the game. This is because the target node is not yet identified in the network; hence, the PLB should consider the probability of correct, wrong, or undecided target node identification before choosing its strategy. In this regard, we use Figure 4.6(a) to show the dependency between a monitoring PLB's strategy and p_k. As shown, the left column corresponds to the player's voting strategy and the right column corresponds to its abstaining strategy. Also, the top row (labeled p_k) refers to the case that the target node is correctly identified in the game,

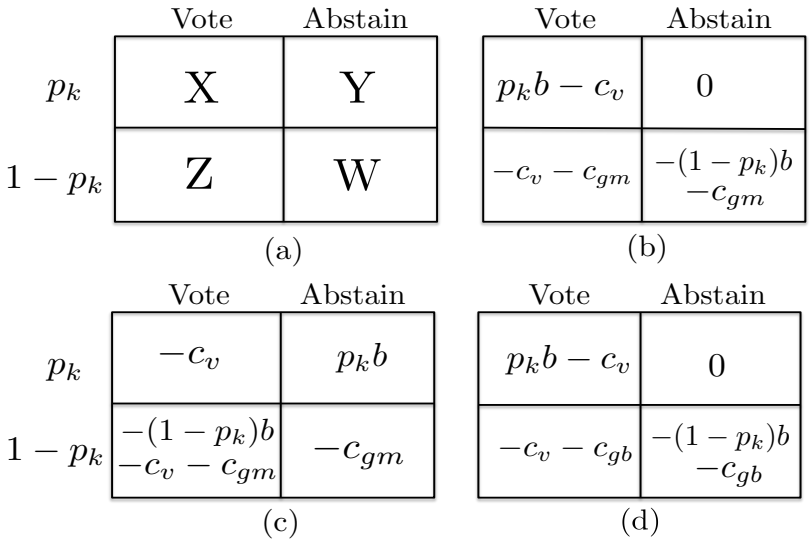


Figure 4.6: Group payoffs: (a) for a monitoring benign player in general, which is then broken down to the following scenarios: (b) malicious target node has attacked a monitoring benign node, (c) malicious target node has not attacked a monitoring benign node, and (d) benign target node versus a monitoring benign node.

and the lower row (labeled $1-p_k$) refers to the case that the target node is not correctly identified in the game. In this figure, X, Y, Z, and W denote payoffs for different possible cases. For example, X in the top left window in Figure 4.6(a) represents the case where the PLB votes and target node is correctly identified in the game. We will design X, Y, Z, and W for different scenarios between the PLT and a monitoring PLB. Using Figure 4.6(a), we define group payoffs for the voting and abstaining strategies of the PLB.

Definition 3: Considering two possible outcomes for target node identification, denoted by p_k and $1-p_k$ in Figure 4.6(a), we define two group payoffs for possible strategies of a monitoring PLB at the k^{th} stage:

$$u_g\,(vote) = p_k\,(X) + (1-p_k)(Z), \tag{4.10}$$

$$u_g\,(abstain) = p_k\,(Y) + (1-p_k)(W), \tag{4.11}$$

where the payoff of each strategy is weighted by the corresponding probabilities.

In the last scenario, since a non-monitoring PLB always abstains from voting regardless of the PLT's strategy, its group payoff depends on other nodes' actions in the game. In what follows, we study payoffs for all the above scenarios to evaluate related $u_{z,g}$ and $v_{z,g}$.

As can be seen by Eqs. (4.10)-(4.11), p_k plays an important role in the payoffs. In this respect, we obtain p_k to evaluate the voting and abstaining strategies of players. It is noteworthy that the value of p_k increases when the PLB votes correctly. We use δ to denote this improvement in p_k.

Lemma 1. *p_k and δ can be written as follows:*

$$p_k = \sum_{i=n_r}^{n_l} \binom{n_l}{i} \left(p_s\right)^i \left(1-p_s\right)^{n_l-i}, \tag{4.12}$$

$$\delta = \binom{n_l}{n_r-1} \left(p_s\right)^{n_r-1} \left(1-p_s\right)^{n_l-(n_r-1)}, \tag{4.13}$$

where $p_s \triangleq \lambda(1-\mu)\alpha P_m$ represents the probability of correct target identification by a remaining node in the game.

Proof. To obtain p_k, note that n_{v1} and n_{v2} votes have already been cast until the k^{th} stage, while there are n_l nodes left in the game. To derive a closed form for p_k, note the following: (i) If n_r ¿ n_l, then $p_k = 0$, which means that the number of left nodes is less than the number of required votes to identify the PLT; (ii) if $n_r = 0$, then $p_k = 1$, which implies that the PLT has been already identified; (iii) p_k directly depends on n_l and their type; and (iv) if n_r is reduced, then p_k will be increased. Taking these points into account, p_k can be written in the form of eq. (4.12), where p_s represents the probability of correct target node identification. For instance, assume $n = 10$, $k = 7$ (i.e., $n_l = 3$), $p_s = 1/3$, and $n_{th} = 4$. Under such assumptions, if $n_{v1} = 0$ (i.e., $n_r = 4$), then equation (4.12) yields $p_k = 0$ because of the first condition. If $n_{v1} = 4$ (i.e., $n_r = 0$), then eq. (4.12) yields $p_k = 1$ because of the second condition. Also, substituting $n_r = 1$ and $n_r = 3$, respectively, yields $p_k = 0.7$ and $p_k \approx 0.04$, which confirm the last condition. We can define $p_s = \lambda(1-\mu)\alpha P_m$, where λ represents the

probability of a remaining node to be in the network and $1-\mu$ is the probability of the remaining node to be benign.

Since δ is defined as the difference that a correct vote can make in p_k, we have

$$\begin{aligned} & \delta = p_k(voting) - p_k(abstaining), \\ \Rightarrow \ & \delta = \sum_{i=n_r-1}^{n_l} \binom{n_l}{i} \left(p_s\right)^i \left(1-p_s\right)^{n_l-i} - \\ & \qquad \sum_{i=n_r}^{n_l} \binom{n_l}{i} \left(p_s\right)^i \left(1-p_s\right)^{n_l-i}, \end{aligned} \tag{4.14}$$

which yields eq. (4.13). □

Scenario I: The monitoring PLB faces the attacking malicious PLT, which relates to the first row of Figure 4.5. The group payoffs in this scenario correspond to $u_{1,g}$, $u_{2,g}$, $v_{1,g}$, and $v_{2,g}$. Figure 4.6(b) shows the voting payoff (i.e., $u_{1,g}$) and the abstaining payoff (i.e., $u_{2,g}$) for a monitoring PLB w.r.t. p_k. As can be seen in Figure 4.6(b), $-c_v$ in the left column (i.e., vote) represents the cost of voting. Also, $-c_{gm}$ in the lower row (i.e., $1-p_k$) denotes the cost of incorrect identification for a malicious target node. The reward for voting in correct target identification (top left window) and the punishment of abstaining in incorrect target identification (bottom right window) are represented by bp_k and $-b(1-p_k)$, respectively. These are proportional to p_k because the player's expected outcome is entangled with the probability of correct target node identification (p_k) in the middle of the game. The reward and the punishment are considered (as incentives) to encourage nodes in cooperation. Using Figure 4.6(b) along with equations (4.10) and (4.11) in definition 3, we have

$$\begin{aligned} & u_{1,g} = p_k \times \big(p_k b - c_v\big) + \big(1-p_k\big) \times \big(-c_v - c_{gm}\big), \\ \Rightarrow \ & u_{1,g} = p_k^2 b - c_v - (1-p_k)c_{gm}, \end{aligned} \tag{4.15}$$

$$\begin{aligned} & u_{2,g} = p_k \times \big(0\big) + \big(1-p_k\big) \times \big(-(1-p_k)b - c_{gm}\big), \\ \Rightarrow \ & u_{2,g} = -(1-p_k)^2 b - (1-p_k)c_{gm}. \end{aligned} \tag{4.16}$$

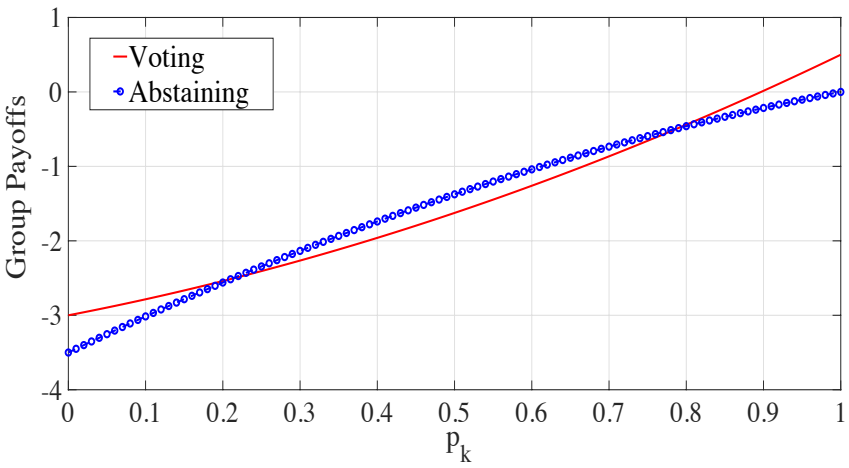


Figure 4.7: Group payoffs for monitoring benign node relative to p_k.

We also define group payoffs for the PLT in this scenario by $(1 - p_k)c_{gm}$, which indicates the inverse proportional relationship between p_k and the gain of the malicious PLT. Hence, we have

$$v_{1,g} = v_{2,g} = (1 - p_k)c_{gm}. \tag{4.17}$$

In order to better understand the impact of a node's group payoff on selecting its strategy, we provide a plot in Figure 4.7 that shows voting payoffs (i.e., $u_{1,g}$) and abstaining payoffs (i.e., $u_{2,g}$) of PLB w.r.t. different p_ks in this scenario. Here, it is assumed that $c_v = 1$, $b = 1.5$, and $c_{gm} = 2$. As can be seen, depending on the value of p_k, voting payoffs can outweigh abstaining payoffs, and vice versa. For example, voting payoffs are dominant for $p_k < 0.2$, which means that the player votes in this interval of p_k. In fact, voting is an attempt by PLB to increase p_k and avoid an incorrect outcome of the game. The main motivation of the player, however, comes from the game's punishment. That is, the cost of voting is lower than the punishment of the game when the malicious target node is not correctly identified. In other words, if $p_k \to 0$, then $-c_v > -(1-p_k)b$ (see lower row of Figure 4.6(a)). Therefore, the player votes not only to increase p_k but also to avoid punishment in the game. The same reasoning can be used for other intervals of p_k, i.e., $p_k > 0.8$ and

$0.2 < p_k < 0.8$, to determine the motivations of the player for voting and abstaining in the game.

Scenario II: The monitoring PLB faces the non-attacking malicious PLT, which relates to the second row of Figure 4.5. The group payoffs in this scenario correspond to $u_{4,g}$, $u_{5,g}$, $v_{4,g}$, and $v_{5,g}$. Figure 4.6(c) shows the voting payoff (i.e., $u_{4,g}$) and the abstaining payoff (i.e., $u_{5,g}$) for a monitoring PLB w.r.t. p_k. Here, a monitoring PLB might unintentionally support a malicious PLT by its vote because the malicious PLT is in a non-attacking state. In this regard, we define a penalty by $-(1-p_k)b$ for voting in an incorrect target identification (bottom left window), and a reward by $p_k b$ for abstaining in a correct target identification (top right window). This prevents the PLB from blind voting (solely to gain the benefit of collaboration) when it did not sense abnormalities from a node. Using Figure 4.6(c) along with equations (4.10) and (4.11) in definition 3, we have

$$\begin{aligned} u_{4,g} &= p_k\big(-c_v\big) + \big(1-p_k\big)\big(-(1-p_k)b - c_v - c_{gm}\big), \\ \Rightarrow u_{4,g} &= -(1-p_k)^2 b - c_v - (1-p_k)c_{gm}, \qquad (4.18) \\ u_{5,g} &= p_k \times \big(p_k b\big) + \big(1-p_k\big) \times \big(-c_{gm}\big), \\ \Rightarrow u_{5,g} &= p_k^2 b - (1-p_k)c_{gm}. \qquad (4.19) \end{aligned}$$

Since the malicious PLT is in a non-attacking mode, we define $v_{4,g} = v_{5,g} = 0$, which implies that the non-attacking PLT does not gain or lose in this scenario.

Scenario III: The monitoring PLB faces a (non-attacking) benign PLT, which relates to the third row of Figure 4.5. The group payoffs in this scenario correspond to $u_{7,g}$, $u_{8,g}$, $v_{7,g}$, and $v_{8,g}$. Figure 4.6(d) shows the voting payoff (i.e., $u_{7,g}$) and the abstaining payoff (i.e., $u_{8,g}$) for a monitoring PLB w.r.t. p_k. The design of payoffs in Figure 4.6(d) is quite similar to scenario I in Figure 4.6(b), where c_{gm} is replaced by c_{gb}. Hence, the reasoning for this scenario follows the same lines as scenario I. Using Figure 4.6(c) along with equations

(4.10) and (4.11) in definition 3, we have

$$u_{7,g} = p_k \times \big(p_k b - c_v\big) + \big(1 - p_k\big) \times \big(- c_v - c_{gb}\big),$$
$$\Rightarrow u_{7,g} = p_k^2 b - c_v - (1 - p_k)c_{gb}, \tag{4.20}$$
$$u_{8,g} = p_k \times \big(0\big) + \big(1 - p_k\big) \times \big(- (1 - p_k)b - c_{gb}\big),$$
$$\Rightarrow u_{8,g} = -(1 - p_k)^2 b - (1 - p_k)c_{gb}. \tag{4.21}$$

Since benign PLT is in a non-attacking mode, we have $v_{7,g} = v_{8,g} = 0$.

Scenario IV: The PLB is in a non-monitoring state. In this case, the PLB abstains from voting, regardless of the PLT's strategy. The group payoffs in this scenario correspond to $u_{3,g}$, $u_{6,g}$, $u_{9,g}$, $v_{3,g}$, $v_{6,g}$, and $v_{9,g}$. To define $u_{3,g}$ and $u_{6,g}$, where the PLT is malicious, we know that the non-monitoring PLB completely relies on other nodes for target node identification. If $p_k = 1$, then the node is not harmed, but if $p_k = 0$, then it gets $-c_{gm}$ as the cost of the incorrect malicious PLT identification. Thus, we define $u_{3,g} = u_{6,g} = -(1-p_k)c_{gm}$, where the node does not impact the group decision while it is affected by other decisions. We can apply a similar reasoning for $u_{9,g}$, with the only difference being that c_{gm} is replaced by c_{gb}, because the PLT is benign, i.e., $u_{9,g} = -(1 - p_k)c_{gb}$. On the other hand, we define $v_{3,g} = (1 - p_k)c_{gm}$, which reflects the inverse proportional relationship between p_k and the gain of the attacking malicious PLT. In the case of $v_{6,g}$ and $v_{9,g}$, the benign PLT does not attack; hence, there is no gain or loss, i.e., $v_{6,g} = v_{9,g} = 0$.

Total payoffs

By designing individual payoffs and group payoffs for each scenario in the game, we can use definition 2 to obtain all payoffs. For example, u_1 is the summation of $u_{1,i}$ (i.e., equation (4.3)) and $u_{1,g}$ (i.e., equation (4.15)). Thus, using equation (4.1) in definition 2, we have

$$u_1 = -c_m + (2\alpha - 1)w + p_k^2 b - c_v - (1 - p_k)c_{gm}, \tag{4.22}$$

We obtain the remaining payoffs in the same fashion, as follows:

$$u_2 = -c_m + (2\alpha - 1)w - (1 - p_k)^2 b - (1 - p_k)c_{gm}, \tag{4.23}$$

$$u_3 = -w - (1 - p_k)c_{gm}, \tag{4.24}$$

$$u_4 = -c_m - \beta w - (1 - p_k)^2 b - c_v - (1 - p_k)c_{gm}, \tag{4.25}$$

$$u_5 = -c_m - \beta w + p_k^2 b - (1 - p_k)c_{gm}, \tag{4.26}$$

$$u_6 = -(1 - p_k)c_{gm}, \tag{4.27}$$

$$u_7 = -c_m - \beta w + p_k^2 b - c_v - (1 - p_k)c_{gb}, \tag{4.28}$$

$$u_8 = -(1 - p_k)^2 b - (1 - p_k)c_{gb} - c_m - \beta w, \tag{4.29}$$

$$u_9 = -(1 - p_k)c_{gb}, \tag{4.30}$$

$$v_1 = v_2 = -c_a - (2\alpha - 1)w + (1 - p_k)c_{gm}, \tag{4.31}$$

$$v_3 = -c_a + w + (1 - p_k)c_{gm}, \tag{4.32}$$

$$v_4 = v_5 = v_6 = v_7 = v_8 = v_9 = 0. \tag{4.33}$$

4.4.4 Variable-Benefit Scheme

So far, we have assumed that the benefit of a correct strategy (b) is constant, irrespective of its impact on the outcome of the game. In principle, a variable benefit could be designed to be commensurate with the impact of k^{th} player's strategy on the target node identification. In this regard, we choose p_k and μ as two important arguments that directly affect the value of a strategy in the game. We categorize these benefits into two cases. The first is the benefit when the PLB has detected an abnormality in the PLT. This implies that the malicious PLT has attacked the PLB, or that the abnormality simply comes from a false alarm. We denote the benefit for a correct strategy in this case by b_1. Here, group payoffs are the same as in Figure 4.6(a), wherein $b = b_1$. The second case is when a monitoring PLB has not detected any abnormalities from the PLT, i.e., whether the PLT is malicious or benign. We denote the benefit for a correct strategy in this case by b_2. Payoff tables for this

case are the same as those in Figure 4.6(b) and Figure 4.6(c), wherein $b = b_2$. By making the PLB indifferent, b_1 and b_2 can be derived.

Lemma 2. *b_1 and b_2 for the k^{th} stage of the game can be obtained as follows:*

$$b_1 = \frac{c_v}{(2p_k^2 - 2p_k + 1)}, \tag{4.34}$$

$$b_2 = \frac{c_v}{(1-2\mu)(2p_k^2 - 2p_k + 1)}. \tag{4.35}$$

Proof. A node must remain indifferent between voting and abstaining for all p_ks and μs. That is,

$$Eu(voting)_z = Eu(abstaining)_z,$$
$$z \triangleq \{attack, not\ attack\}, \tag{4.36}$$

where $Eu(.)$ denotes expected utility function, and z is the strategy of PLT. Applying eq. (4.36) for b_1 (Figure 4.6(a)), and assuming small values of β, we obtain

$$p_k(p_k b_1 - c_v) + (1-p_k)(-c_v - c_{gm})$$
$$= (1-p_k)[-(1-p_k)b_1 - c_{gm}]. \tag{4.37}$$

Simplifying eq. (4.37) yields eq. (4.34). Eq. (4.35) can be obtained in the same fashion using Figs. 4.6(b)-(c). □

Assuming $c_v = 1$, the value of b_1 and b_2 versus p_k and μ are shown in Figure 4.8. As can be seen in Figure 4.8(a), when $p_k = 0.5$, the highest b_1 occurs, and when $p_k = 0$ or $p_k = 1$, the lowest b_1 occurs, underlining the fact that the highest benefit is rewarded for a PLB that faces the highest uncertainty of p_k (similar to gambling!). On the other hand, as shown in Figure 4.8(b), when μ (the portion of malicious nodes) grows, the value of benefits also increases. This is not surprising because as μ increases, benign nodes should be more

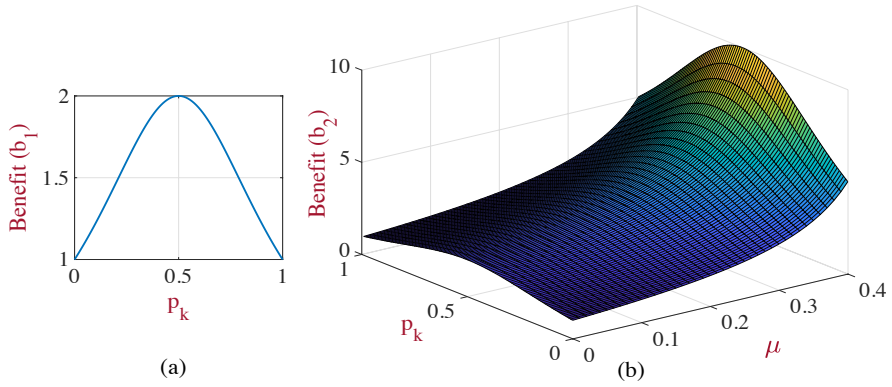


Figure 4.8: Benefits with regard to probability of successful target identification in k^{th} stage (p_k) and portion of malicious nodes (μ) in network.

motivated for participation to reveal the identity of a target node before malicious nodes determine the game's result with their votes.

Here, we study the impact of b_1 on group payoffs. The study of group payoffs with b_2 can be done in the same fashion. Figure 4.9 shows group payoffs with $b = b_1$, where each subplot corresponds to a window in Figure 4.6(b). For example, the top left subplot in Figure 4.9 refers to $p_k b - c_v$ in Figure 4.6(b), where $b = b_1$. In Figure 4.9, we assume that $c_v = 1$ and $c_{gm} = 4$. As can be seen, payoffs in the upper row dominate over those in the lower row. This is because the upper payoffs show successful target node identification, while the lower payoffs indicate the unsuccessful counterpart. To understand this better, let us study the trend of graphs for a few p_ks. First, let $p_k = 0.25$, which yields $b_1 = 1.6$ from eq. (4.34). Under this assumption, if the game is to successfully identify the target node (upper row in Figure 4.9), then PL2 collects a higher payoff by abstaining ($0 > -0.6$). However, if the game becomes unsuccessful (lower row), then PL2 should have voted in the game ($-5 > -5.2$). Next, assume that $p_k = 0.75$. Interestingly, we obtain the same benefit (i.e., $b_1 = 1.6$). In this case, however, the strategy of PL2 will be the opposite. That is, PL2 is willing to vote if it thinks the game will be successful ($0.2 > 0$), while it abstains if it thinks the game will be unsuccessful ($-.4.4 > -5$). This example helps us summarize the following intuitions for the case of adaptive benefit: (i) There is no pure strategy for PL2 during the game w.r.t. a specific p_k or b; and (ii) $p_k = 0.5$ is the only point where

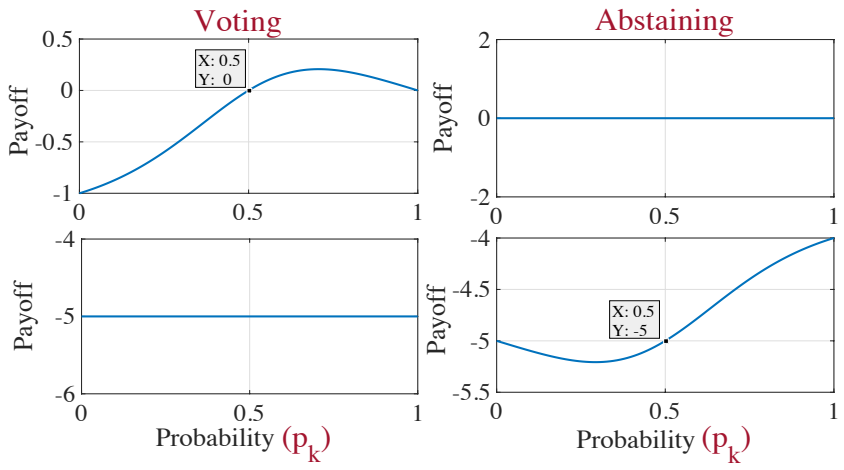


Figure 4.9: Expected group payoffs for scenario I with variable benefits.

PL2 becomes completely indifferent between voting and abstaining, regardless of the result of the game (see $p_k = 0.5$ in Figure 4.9). This is the place where our design offers the highest benefit (see Figure 4.8) in order to encourage PL2 to participate, hence, increase p_k.

4.5 Equilibrium Analysis

The objective of the players is to maximize their payoffs in the game. In this regard, we obtain possible equilibrium points using a Bayesian game to better understand the behavior of the players. In particular, we obtain the best strategies of benign players to identify a malicious node, while we find the maximum level of aggressiveness for malicious nodes without being identified. In this respect, we use the interactions between a PLB and a PLT, as illustrated in Figure 4.5. Let us quickly summarize the possible strategies of the players. A non-monitoring PLB has one pure strategy: abstaining. A monitoring PLB has two strategies: voting or abstaining. A benign PLT has one pure strategy: to not attack. Finally, a malicious PLT could choose two strategies: to attack or not to attack. Depending on game parameters, a pure-strategy BNE may or may not exist. This is addressed in the following theorem.

Theorem 6. *Given μ and P_m, if $c_{gm} \geq c_a + (2\alpha - 1)\, w + \delta c_{gm}$, and $b > 2\, c_v$, then the game has one pure-strategy BNE. A malicious node attacks and a monitoring benign node votes in the game.*

Proof. We first derive combined payoffs for two types of PLT (malicious and benign) and the PLB. Then, using strictly dominated strategies, we prove the theorem w.r.t. the conditions.

Figure 4.10 shows the combination of payoffs. The first and the second element of each column (e.g., $\{A, NA\}$) indicate the strategy of a malicious PLT and a benign PLT. For instance, $\{A, NA\}$ represents the attacking (A) and the non-attacking (NA) strategies from PLT. In addition, ϕ_is and ν_is denote expected payoffs for the PLT and the PLB, respectively. For instance, ν_1 is obtained as follows:

$$\nu_1 = \mu P_m u_1 + (1 - \mu) P_m u_7, \tag{4.38}$$

where u_1 and u_7 are obtained in eqs. (4.22) and (4.28), respectively. Based on the values of ϕ_is, if $c_{gm} \geq c_a + (2\alpha - 1)\, w + \delta c_{gm}$, then the left column in Figure 4.10 ($\{A, NA\}$) strictly dominates the right column ($\{NA, NA\}$). On the other hand, we need to show that $\nu_1 > \nu_3$ to obtain pure-strategy BNE. Substituting the equations of ν_1 and ν_3 yields

$$\begin{aligned} &(1 - P_m)\mu w + (1 - P_m)(1 - \mu)(1 - p_k) c_{gb} + (1 - P_m) \times \\ &\mu (1 - p_k) c_{gm} + P_m [(2p_k^2 - 2p_k + 1)\, b - c_v] > 0 \end{aligned} \tag{4.39}$$

As can be seen, the first three items in eq. (4.39) are equal to or greater than zero. Hence, it suffices to say that the last term is positive. This happens only if $b > \frac{c_v}{(2p_k^2 - 2p_k + 1)}$. This inequality, however, must hold for all p_ks, which means that the right-hand side of the inequality must be maximized. This yields $p_k = 0.5$, which implies $b > 2\, c_v$. □

Remark 1: When the damage caused by incorrect malicious node identification (c_{gm}) is higher than the summation of PLT's cost of attack (c_a), PLT's individual loss against a

PLB \ PLT	{A, NA}	{NA, NA}
Vote	$\nu_1,\ \phi_1$	$\nu_2,\ \phi_2$
Abstain	$\nu_3,\ \phi_3$	$\nu_4,\ \phi_4$

Figure 4.10: Expected payoffs for combined types of PLT and the PLB.

monitoring PLB, i.e., $(2\alpha - 1)\, w$, (see eq. (4.4)), and the impact of PLB's vote in the game (δc_{gm}), then a malicious node attacks a benign node. When the benefit of voting for a monitoring PLB is more than twice of its cost, then the benign node chooses to vote.

The pure-strategy BNE, as seen above, holds only under certain conditions on the parameters. Our game is a finite strategic-form game. Hence, a mixed-strategy BNE can be applied to gain a broader perspective for the game analysis. In this regard, to determine each player's indifference strategy, we define q as the probability of attack for a malicious PLT, and s as the probability of voting for a monitoring PLB.

Theorem 7. *Given μ and P_m, the game defined in section III has a mixed-strategy BNE, which is as follows:*

- *Malicious node attacks with a probability of q^*, which is*

$$q^* = \frac{q_1 + q_2 + ... + q_n}{n}, \tag{4.40}$$

where q_k, $k = 1, ..., n$, is the probability of attack for the k_{th} node in the game

$$q_k = \frac{A_k}{B_k}, \tag{4.41}$$

$$\begin{aligned} A_k &= \mu\big(1 + P_m\big)\big(2p_k^2 - 2p_k + 1\big)b + \big(1 - P_m\big) \\ &\quad \times \big(c_m + \beta w\big) + c_v - p_k^2 b - P_m\big(1 - p_k\big)^2 b, \\ B_k &= \mu\big(1 + P_m\big)\big(2p_k^2 - 2p_k + 1\big)b \end{aligned}$$

$$+ \mu\big(1 - P_m\big)(2\alpha + \beta)w.$$

- *Monitoring benign node votes with probability of s^*, which is equal to*

$$s^* = \frac{c_a + (2\alpha P_m - 1)w - c_{gm}}{(1 - P_m)[-c_a + (1 - 2\alpha)w + c_{gm}] - \delta c_{gm}}. \tag{4.42}$$

Proof. To obtain q^*, we first equalize the expected utilities for voting and abstaining to obtain q_k. Then, we take an average over all possible values of the p_ks to get eq. (4.40). In this way

$$Eu\left[voting\right] = Eu\left[abstaining\right] \tag{4.43}$$

where,

$$Eu\left[voting\right] = \mu q u_1 + \mu\big(1 - q\big)u_4 + \big(1 - \mu\big)u_7, \tag{4.44}$$

$$\begin{aligned} Eu\big[abstaining\big] = {} & \mu\, q\, P_m u_2 + \mu\, q\big(1 - P_m\big)u_3 + \mu\,\big(1- \\ & q\big)P_m u_5 + \mu\big(1 - q\big)\big(1 - P_m\big)u_6 + \big(1- \\ & \mu\big)P_m u_8 + \big(1 - \mu\big)\big(1 - P_m\big)u_9. \end{aligned} \tag{4.45}$$

Substituting eqs. (4.22), (4.25), and (4.28) into eq. (4.44), and eqs. (4.23), (4.24), (4.26), (4.27), (4.29), and (4.30) into eq. (4.45), and then substituting eqs. (4.44) and (4.45) in eq. (4.43) yields eq. (4.41). Since the malicious PLT might attack the neighboring nodes regardless of their stage in the game, we take an average over all values of q_ks, which yields eq. (4.40).

To calculate s^*, we can equalize the expected utilities of attack and not attack from the PLT, hence obtaining

$$\mu\, s\, v_1 + P_m\, \mu\, (1-s)\, v_2 + (1-P_m)\, \mu\, v_3 = 0. \tag{4.46}$$

Plugging eqs. (4.31) and (4.32) back into eq. (4.46) yields eq. (4.42). □

Note that the mixed-strategy provides general equilibrium points w.r.t. different parameters. In a special case, if all nodes monitor their neighbors, i.e., $P_m = 1$, then an upper bound for the benefit and a lower bound for the detection rate can be derived using eqs. (4.41) and (4.42), respectively.

Corollary 1. *In Theorem 2, if $P_m = 1$, then*

$$b < \frac{c_v}{(1-2\mu)(2p_k^2 - 2p_k + 1)}, \tag{4.47}$$

$$\alpha > \frac{w - c_a + c_{gm}(1-\delta)}{2w}. \tag{4.48}$$

From eq. (4.47), it can be seen that as $\mu \to \frac{1}{2}$, the upper bound increases. This allows network designers to select higher values of benefit in environments where the probability of a malicious PLT is higher. On the other hand, eq. (4.48) implies that a monitoring system must have a minimum true positive rate in order to make a malicious node indifferent in the game.

Corollary 2. *In Theorem 2, if $P_m = 1$, and benefits are designed using b_1 and b_2 in Eqs. (4.34) and (4.35), then the probability of attack by a malicious node will be zero, i.e., $q^* = 0$.*

While the result of this corollary is the most desirable outcome for every game designer, we should note that achieving it might require a strong assumption ($P_m = 1$) and

a complex system for a beneficial design, because it requires all monitoring nodes in an ephemeral network that adapt their benefits in each stage according to p_k.

In the next section, we evaluate the performance of the game in order to obtain a better picture of the above analysis.

4.6 Numerical Results

To evaluate our analysis, we assume that 40 nodes can run the game in an area that is 625 m $\times$ 625 m (normal density $\approx 100\frac{\text{nodes}}{km^2}$ in [115]). Since the analysis is probabilistic, we run 100 iterations for each simulation. Then, we take an average of the results with 95% confidence interval. The default game parameters are as follows:

- Monitoring system parameters: $\alpha = 0.95$, $\beta = 0.05$,
- Probabilities: $P_m = 0.75$, $\mu = 0.2$, and $q = 0.4$,
- Costs and benefits: $c_{gb} = c_{gm} = 4$, $w = 4$, $b = 3$, $c_m = c_a = c_v = 1$.

If we change these parameters to better explain a scenario, then we will explicitly mention it. Nevertheless, we describe the theoretical results in three subsections. Initially, we study the impact of incentives (in particular, b) on correct, wrong, and undecided target node identification. Then, we focus on the behavior of malicious nodes w.r.t. their portion and aggressiveness in the network. This section also includes a comparison among different n_{th}s. Finally, we compare our work with scenarios where the uncertainties discussed in this chapter have not been considered, e.g., [103, 13, 100]. In all these cases, we evaluate and compare the percentage of target node identification for different sets of given parameters that lead to different equilibria for the game.

4.6.1 Impact of incentives

Figure 4.11 illustrates the percentage of target node identification versus b. Here, it is assumed that $q = 0.7$. As shown, this figure can be categorized into four different

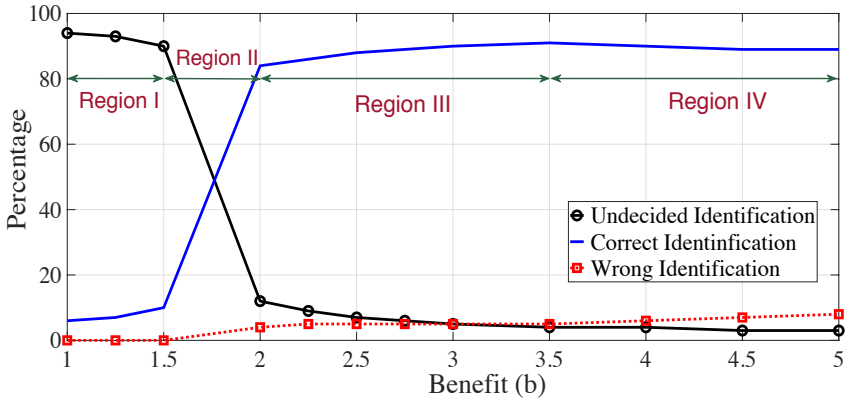


Figure 4.11: Game outcomes versus benefit variations.

regions. In region I, the percentage of undecided target identification outweighs correct and wrong identifications for a simple reason: the benefit is not large enough to persuade nodes to participate in the game. Region II, however, illustrates a drastic reduction of undecided identification. This indicates that voting payoffs become larger in comparison to abstaining payoffs. In addition, correct identification dominates over wrong identification, which is the result of the following: (i) benign nodes with high monitoring and detection rates (i.e., $P_m = 0.75$ and $\alpha = 0.95$), and (ii) malicious nodes with a high level of aggressiveness (i.e., $q = 0.7$). Region III shows a slight increase in correct identification and a decrease in undecided identification because of lower payoffs for abstaining from the game. The increase of wrong identification over undecided identification is remarkable in region IV. Wrong votes in this region mainly come from highly encouraged benign nodes that have not been attacked by a malicious target node. In other words, since voting payoffs are significantly larger than abstaining payoffs (i.e., $u_4 > u_5$ and $u_7 > u_8$), a benign node votes in favor of a non-attacking target node. This observation reveals that persuading every node to vote by applying the leverage of benefit does not necessarily lead to a better outcome. Taking all regions into consideration, region III indicates the best option for the benefit design.

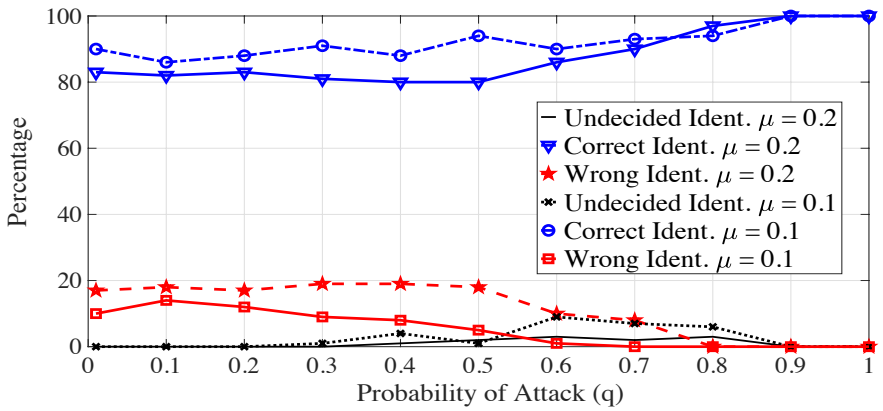


Figure 4.12: Impact of portion of malicious nodes (μ) and probability of attack (q) on identification results.

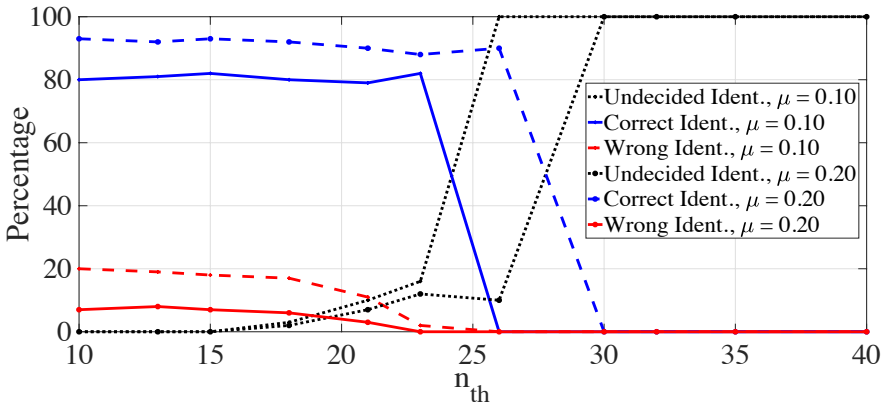


Figure 4.13: Impact of required votes, n_{th}, on target identification.

4.6.2 Impact of malicious nodes and n_{th}

Figure 4.12 shows the percentage of target identification w.r.t. the portion of malicious nodes and their probability of attack (q) in the network. As shown, when q increases, correct identification generally increases, which confirms that aggressive attackers can be more easily identified. However, wrong identification is reduced after a certain value of q; for example, $q = 0.1$ for $\mu = 0.1$. When the number of malicious nodes increases in the network, this decreasing trend starts at higher values of q; for instance, $q = 0.4$ for $\mu = 0.2$. This reveals that malicious nodes become more aggressive when their number increases in the network.

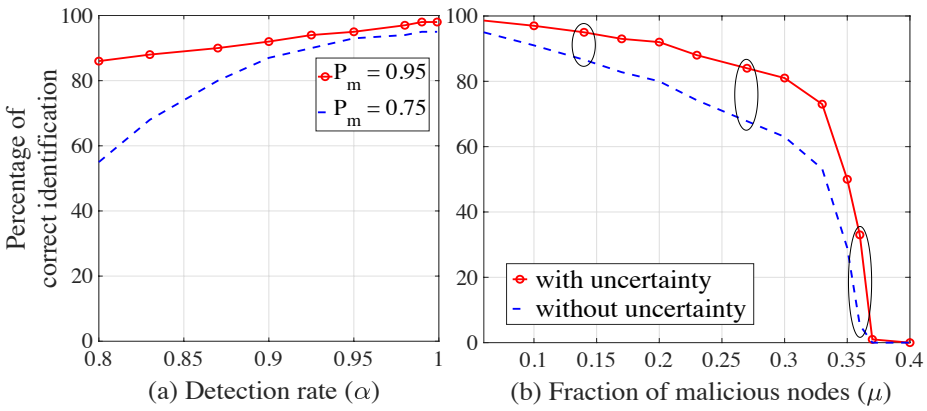


Figure 4.14: Impact of game uncertainties relative: (a) detection rate, and (b) correct identification rate.

Figure 4.13 depicts identification results w.r.t. the variation of n_{th} and μ. As n_{th} grows larger, a turning point occurs, whereby undecided identification prevails over correct and wrong identifications. The reason is that a higher number of nodes must participate to make an outcome of identification. For example, consider plots for $\mu = 0.2$, where $n_{th} = 23$ is the turning point. This number comes from the fact that out of 40 neighboring nodes, on average, 8 of them are malicious nodes (40×0.2). Also, from the remaining 32 benign nodes, those that are in the monitoring state, i.e., 24 (32×0.75), might vote depending on their payoffs. If $n_{th} = 23$, then almost all such nodes must participate to avoid undecided target identification. A designer can adjust the n_{th} w.r.t. a range of μs to obtain an acceptable correct identification rate.

4.6.3 Comparison

In this section, we compare our work with the scenarios where some of the explained uncertainties have not been considered (e.g.[103, 13, 100]). It is worth mentioning that the comparison is limited to highlighting uncertainties in those scenarios. This is because the nature of their games and objectives are slightly different. However, this comparison provides us with insight into the effect of incomplete information at nodes on the outcome of a local voting game.

Figure 4.14(a) shows the impact of the true positive detection rate (α) on the correct target identification. As can be seen, it is essential for nodes to have high values of α in order to gain high correct target identification. The values of α become more important when fewer benign nodes monitor their neighbors (i.e., smaller P_m). Figure 4.14(b) indicates a comparison between a design with and without uncertainties in the local voting game. In particular, we assume that a design without uncertainty has the following parameters: $\alpha = 1, \beta = 0$, and $q = 1$. As shown, the difference between graphs is the growth of μ. This is because a player without uncertainty considers a non-attacking malicious node as a benign node and votes for it. On the other hand, the proposed design prevents benign nodes from voting when they are unsure about the strategy of malicious nodes. In both scenarios, when μ goes beyond a threshold, here 0.3, correct target identification is significantly reduced. This comes from higher payoffs for abstaining in comparison to voting. Interpreted differently, benign nodes are unwilling to cooperate in a game in which a high portion of participants are malicious.

4.7 Summary

In this chapter, we have provided a game-theoretic approach to identify malicious nodes in ephemeral networks, where central stations are not available. In particular, we have studied the strategies of nodes in a local voting-based game using a Bayesian game, in which nodes have incomplete information about the accuracy of their monitoring systems, the type of neighbors (benign or malicious), and the outcome of the game. By offering incentives in expected utilities, we have provided encouragements for game participation with the aim of improving correct node identification. We have derived possible Bayesian Nash equilibrium (BNE) points and mixed-strategy BNE points to study the best strategies of players in the game. Simulation results have shown the impact of different parameters, such as participation benefits and detection rate, on the identification of malicious nodes.

CHAPTER 5

FUTURE WORK: APPLICATION OF LEARNING METHODS FOR SECURITY ANALYSIS OF CYBER-PHYSICAL SYSTEMS

Recent developments and a large amount of data are motivating many researchers to employ learning models as a powerful toolbox to improve the performance of cyber-physical systems. However, it is crucial to select a suitable learning method based on the specific needs of an application. This dissertation continues by introducing the application of *deep reinforcement learning* to CPSs. In this respect, a brief framework is provided as future work in which smartphones and the Internet of Things (IoT) devices such as sensors, cameras, and wearable devices can employ learning methods to improve their security and performance in a CPS. In particular, offloading tasks from mobile devices to Mobile Edge Computing (MEC) servers in the presence of malicious nodes is the main focus. Mobile devices as the lower layer of a network usually face limited power, memory, and computational resources, while MEC servers are the higher level of the network and have sufficient resources to store data and accomplish tasks. Hence, there is an opportunity for mobile devices to offload their tasks onto a MEC server. However, the existence of malicious nodes in the network can compromise the solution. Therefore, it is essential for mobile devices to have defense strategies against malicious nodes to improve their security and performance.

The organization of this chapter is as follows. Section 5.1 provides an overview of deep reinforcement learning and the deep Q-learning algorithm. Section 5.2. briefly describes a secure and efficient model for offloading the tasks of mobile devices to access points. In this section, an optimization problem is set between mobile IoT devices and a MEC, where the goal of the objective function is to minimize energy and the processing delay of nodes in the network.

5.1 Deep Reinforcement Learning

First, the Markov decision process (MDP) is described in order to explain reinforcement learning. Then, using Q-learning, a typical algorithm of deep reinforcement learning that can be extended for an application is introduced.

5.1.1 Markov Decision Process

The MDP provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker (an agent) [116]. The relation between an agent and the environment is shown in Figure 5.1. The Markov random process is usually described by a 4-tuple (S, A, R, P_a), where S denotes a set of finite states, A denotes a set of finite actions, R represents a set of rewards, and P_a shows the probability of transition from one state to another state caused by action $a \in A$. A policy, $\pi(s)$, maps states to actions for an agent, i.e., $\pi : S \rightarrow A$. The goal of an MDP is to find a policy that determines the best action for an agent when it is in state s. To find an optimal policy, $\pi^*(s)$, the value of each state under the policy π should be obtained. This is called the state-value function and is denoted by $V^\pi(s)$, which indicates how good any given state is for an agent with a policy π. $V^\pi(s)$ can be written as

$$V^\pi(s) = E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|\, S_t = s\right], \tag{5.1}$$

where γ is a discount factor, and t refers to a time step that an action is taken. The expected value (sum) represents the discounted sum of rewards over (potentially) infinite horizon given state s. The optimal policy can be derived as follows:

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P_a(s, s')\ V^\pi(s) \tag{5.2}$$

This formula returns an optimal value that can come from different actions. In addition, it is important to know the transition probability between two states, P. An action-value

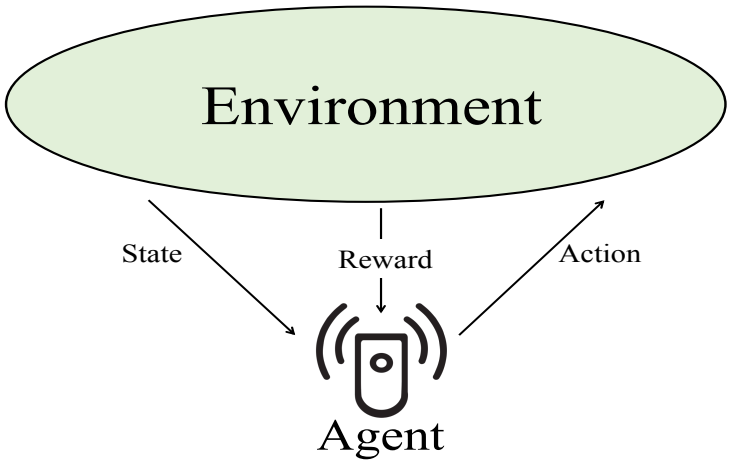


Figure 5.1: Relation between agent and environment in MDP.

function, which is also called a Q-function, can be defined to determine a set of optimal actions. Quite similar to $V^{\pi}(s)$, the Q function is defined as

$$Q^{\pi}(s,a) = E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|\, S_t = s, A_t = a\right]. \tag{5.3}$$

Using the action-value function, the optimal policy can be found without knowing P. Q-learning is a method that can be used to accomplish this task.

5.1.2 Q-Learning

Q-learning is a model-free (as opposed to model-based) reinforcement learning algorithm that learns an optimal policy without using the transition probability distribution associated with the Markov decision process. The optimal Q-function function is

$$Q^*(s,a) = \arg\max_{\pi} \; Q^{\pi}(s,a). \tag{5.4}$$

Q^* provides the maximum expected return achievable by all policies for each action-state pair. To obtain Q^*, the well-known Bellman equation can be used, and written as

$$Q^*(s,a) = E\left[R_{t+1} + \gamma \max_{a'} \; Q^*(s',a')\right] \tag{5.5}$$

The Q-learning algorithm iteratively updates the value of Q using equation (5.5) to find the optimal action-state value, i.e., Q^*. Once it is obtained, then the optimal policy, i.e., $\pi^*(s)$, can be derived from the right side of equation (5.4). The way that the Q value is updated is crucial to find Q^*; hence, the exploration and exploitation of actions in the environment will be explained in section 5.1.3.

5.1.3 Exploration vs. Exploitation

Exploration is nothing more than the act of exploring the environment to obtain information about it. Exploitation, however, is the act of exploiting the knowledge that an agent has already received from the environment with the aim of maximizing the return. To strike a balance between exploration and exploitation, the ϵ-greedy algorithm could be used to help an agent decide whether to explore or to exploit the environment. The exploration rate is defined with the ϵ parameter in an algorithm. This ϵ parameter represents a probability that the agent explores the environment. As the agent starts exploring the environment, the value of ϵ decays accordingly.

5.1.4 Loss Function and Learning Rate

Loss function shows the difference between an optimal action-state value, i.e. Q^*, and the value of Q. In other words,

$$L \;=\; Q^*(s,a) - Q(s,a). \tag{5.6}$$

In order to update Q to reach Q^*, the learning rate, α, which is a value between 0 and 1 could be used. This parameter actually shows how quickly the agent moves from old Q values to new values. For instance, if $\alpha = 1$, then $Q_{updated} = Q_{new}$. Taking the learning rate into consideration, Q_{new} can be calculated as

$$Q_{new}(s,a) \;=\; \big(1-\alpha\big)\, Q(s,a) \;+\; \alpha\,\big(R_{t+1} + \gamma \max_{a'}\; Q^*(s',a')\big). \tag{5.7}$$

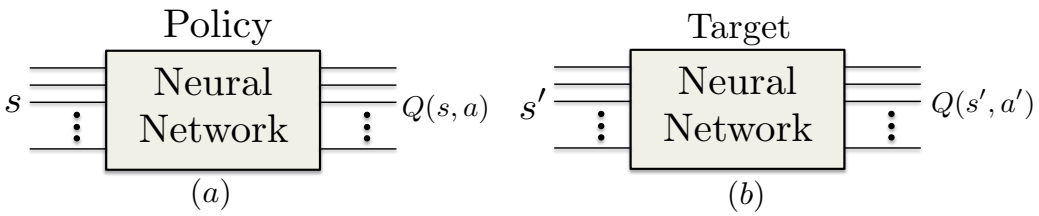


Figure 5.2: Neural networks: (a) policy, and (b) target.

5.1.5 Deep Q-Learning Algorithm

Deep Q-learning employs deep neural networks (NNs) to estimate Q values for each action-state pair in a given environment. The goal is to use DNNs in order to approximate optimal Qs, hence the optimal policy, by minimizing the loss function. The weights of the network iteratively get updated using gradient descent back propagation until the loss function reaches its minimum value. The policy neural network is shown in Figure 5.2(a).

Replay Memory

At time step t, the agent's experience is defined as a tuple, e.g., $e_t = (s_t,\, a_t,\, r_{t+1},\, s_{t+1})$. In replay memory, some of these experiences are stored and will be used as input data to train the neural network. It is worth mentioning that the samples are randomly selected to break the correlation between consecutive samples. Otherwise, the samples are highly correlated, which leads to inefficient learning.

Target Neural Network

The problem with having only one network, i.e., policy neural network, is that only one network is used to obtain both $Q\,(s,a)$ and $Q\,(s',a')$. This means that the weights of the network are updated to obtain a new Q, but Q^* also gets updated through the same network that makes the optimization, similar to chasing its own tail. To solve this problem, s' is fed to a completely different network, as shown in Figure 5.2(b), referred to as a target neural network. In this network the weights are frozen to the original policy network weights, and the weights are updated periodically. Details of a deep reinforcement learning are shown in algorithm 1.

Algorithm 1: Deep Reinforcement Learning

```
Initialize "replay memory" capacity
Initialize "policy network" and "target network" with random weights
for episode do
    Initialize starting state s
    for time step do
        Select an action a (via exploration versus exploitation)
        Execute a in the environment and observe reward r and the next state s'
        Store experience in replay memory
        Sample random batch from random memory
        Preprocess states for the selected batch
        Feed the policy network with the batch
        Feed the target network with the s'
        Calc. loss by Q_{policy network} − Q_{target network}
        Gradient descend update weights in policy network to minimize the error
        After a period of time the weights in the target node are updated.
    end
end
```

5.2 Secure Offloading Tasks from Mobile Devices to Mobile Edge Computing Servers

5.2.1 Motivation

With emerging computation-intensive and energy-consuming applications in mobile devices such as interactive games and augmented reality, solving the problem of battery power and computing capacity is of paramount importance [117]. One viable solution is to use cloud-based services whereby mobile devices can offload their data and computational tasks to MEC servers. This will help those devices realize larger data storage, save energy, and increase the processing speed. However, security is still a bottleneck for developing offloading tasks to servers, because edge devices can be compromised by many attacks. In particular, attackers can impact the exchange of information between legitimate users and MEC servers by conducting denial of service (DoS) attacks within a certain communication range or by sending bogus requests to the server. These attacks may result in reducing the energy efficiency and increasing the processing delay in the network.

With limited energy, communication range, and computational and memory resources, mobile devices should be able to apply some defense protocols to be able to protect themselves agains different attacks. Each node can optimize its defense strategy and select key parameters in the protocols, which are very challenging in wireless networks because the dynamic model of the network and the model of attack are difficult to estimate [118]. This dilemma can be solved by reinforcement learning, wherein each agent can obtain its best strategy by trial and error. In other words, agents use their future reward feedback from the environment to adjust their policies for the best long-term strategy.

5.2.2 Problem Definition and Formulation

Assume that there are a number of mobile devices in a network that could be either benign or malicious, as shown in Figure 5.3. Malicious nodes are able to send bogus requests and conduct DoS attacks. It is reasonable to assume that malicious nodes are not known in the network (the case of known malicious nodes in the network is trivial simply because the MEC server does not accept the offloading requests of those nodes). On the other hand, benign users and the MEC server are able to monitor the environment and, through the measurements of receiving signals, try to adjust their strategies to achieve the best performance. Knowing that both malicious and benign devices have power limitations and request constraints, it is necessary to set an optimization problem in order to achieve the least delay and energy consumption for the benign agents.

In order to set up this optimization formula, the constraints of the problem must be considered. A crucial limitation of both benign and malicious nodes is the number of times that each device can request for offloading tasks to the MEC server. Within a certain number of time slots, as denoted by T, each device can request N number of times for offloading tasks to the server. This can be represented by

$$\alpha_M \triangleq \{\alpha_1, \alpha_2, ..., \alpha_T\}, \tag{5.8}$$

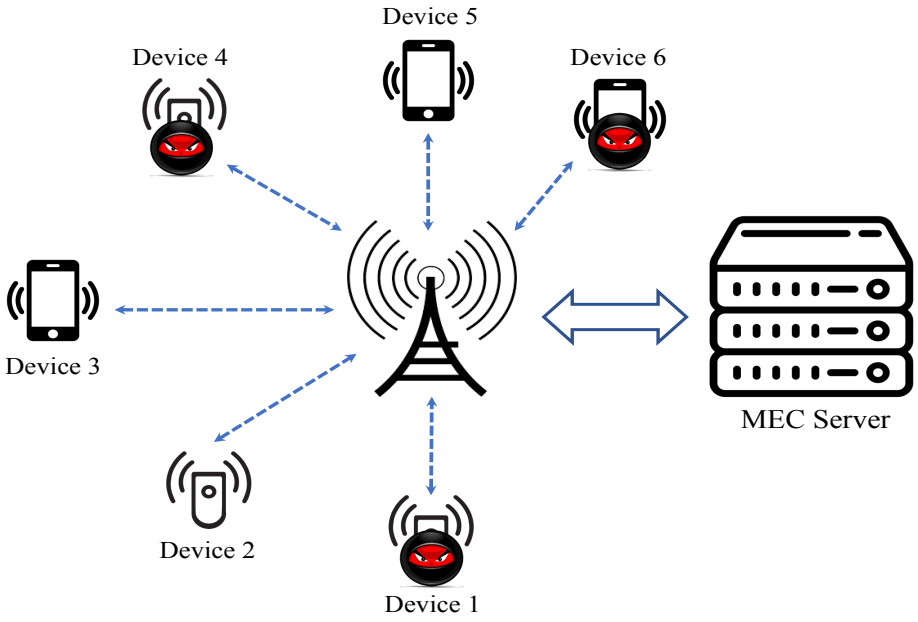


Figure 5.3: Mobile devices offload tasks on MEC servers in presence of malicious nodes.

$$\beta_B \triangleq \{\beta_1, \beta_2, ..., \beta_T\}, \tag{5.9}$$

where $\alpha_i = 1$ and $\beta_i = 1$ denote the offloading request in the i-th time slot for a malicious device and a benign device, respectively. Therefore, the following constraints are for sending requests to the MEC server:

$$\sum_{k=1}^{T} \alpha_k \leq N, \tag{5.10}$$

$$\sum_{k=1}^{T} \beta_k \leq N. \tag{5.11}$$

Energy constraint is another limitation of mobile devices because they should efficiently use their battery power within the period of T time slots. In the same fashion of defining request constraints, the power limitation of mobile devices can be written as follows:

$$\theta_M \triangleq \{\theta_1, \theta_2, ..., \theta_T\}, \text{ and } \omega_B \triangleq \{\omega_1, \omega_2, ..., \omega_T\}, \tag{5.12}$$

$$\sum_{k=1}^{T} \theta_k \leq \Theta, \text{ and } \sum_{k=1}^{T} \omega_k \leq \Omega, \tag{5.13}$$

where Θ and Ω represent power constraints over T time slots for the malicious and benign devices, respectively. Also, θ_i and ω_i denote the power usage of the malicious device and the benign device in the i-th time slot, respectively.

The next step is to define a cost function that includes the limitation of offloading requests for devices, their battery power, and total processing time. This function can be represented by J, and $\alpha_M, \theta_M, \beta_B$, and ω_B should be its main arguments because of constraints. This cost function operates as the objective function of an optimization problem. Two optimization problems are defined: one for benign nodes and the MEC server, and the other for malicious nodes in the network. The goal of the first optimization problem is to minimize the overall cost of offloading tasks for benign nodes in the network, while the second problem aims to maximize this cost. The exact mathematical formula for the cost function has not been obtained, but the optimization problems can be written as follows:

$$\begin{aligned} \min_{\beta_B,\ \omega_B} \quad & J(\beta_B, \omega_B) \\ \text{s.t.} \quad & \sum_{k=1}^{T} \omega_k \leq \Omega, \\ & \sum_{k=1}^{T} \beta_k \leq N, \end{aligned} \tag{5.14}$$

Similarly, the optimization problem for malicious nodes could be written as follows:

$$\begin{aligned} \max_{\alpha_M,\ \theta_M} \quad & J(\alpha_M, \theta_M) \\ \text{s.t.} \quad & \sum_{k=1}^{T} \theta_k \leq \Theta, \\ & \sum_{k=1}^{T} \alpha_k \leq N, \end{aligned} \tag{5.15}$$

Problems (5.14) and (5.15) can be solved by finding optimal values of offloading decision vectors α_M and β_B, and battery power vectors θ_M and ω_B. However, since the vectors of α_M and β_B are binary, the feasible set and objective functional can be non-convex [119]. Also, the size of these problems can quickly grow if the number of time slots

increases. Therefore, the result may be a non-convex problem that is NP hard to solve, extended from the knapsack problem [120]. Instead of solving the NP hard problems by conventional methods, the deep reinforcement learning could be employed to provide an acceptable approximation with less complexity.

In order to use the DRL, its three key elements, i.e., state, action, and reward must be defined for the aforementioned problem. One definition could be as follows:

- State: The state of the system consists of the available capacity of storage and computational resources at MEC, and the overall cost of the problem, J.
- Action: The action of the system consists of offloading tasks from mobile devices, i.e., α_M and β_B, and the amount of power they spend for offloading, i.e., θ_M and ω_B.
- Reward: Devices receive a penalty or a reward according to their actions in different states. The reward function should be related to the objective functions of equations (5.14) and (5.15), which means a higher reward for actions of lower and higher costs for benign nodes and malicious nodes, respectively.

With these terms, the DRL can be used to update the Q-learning function in equation (5.7) to find the the best cost functions.

REFERENCES

REFERENCES

[1] Andersson, G., Donalek, P., Farmer, R., Hatziargyriou, N., Kamwa, I., Kundur, P., Martins, N., Paserba, J., Pourbeik, P., Sanchez-Gasca, J., et al., "Causes of the 2003 major grid blackouts in North America and Europe, and recommended means to improve system dynamic performance," *IEEE Transactions on Power Systems*, Vol. 20, No. 4, 2005, pp. 1922–1928.

[2] Bienstock, D., *Electrical transmission system cascades and vulnerability: an operations research viewpoint*, Vol. 22, SIAM, 2015.

[3] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., and Madden, S., "CarTel: a distributed mobile sensor computing system," Proceedings of the 4th international conference on Embedded networked sensor systems, 2006, pp. 125–138.

[4] Amoozadeh, M., *Towards robust and secure collaborative driving and interactive traffic intersections*, University of California, Davis, 2018.

[5] Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E., and Havlin, S., "Catastrophic cascade of failures in interdependent networks," *Nature*, Vol. 464, No. 7291, 2010, pp. 1025.

[6] Petit, J. and Shladover, S. E., "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 2, April 2015, pp. 546–556.

[7] Parkinson, S., Ward, P., Wilson, K., and Miller, J., "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 11, Nov 2017, pp. 2898–2915.

[8] Yan, C., Xu, W., and Liu, J., "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, Vol. 24, 2016.

[9] Stamatis, D. H., *Failure mode and effect analysis: FMEA from theory to execution*, ASQ Quality Press, 2003.

[10] Frigault, M., Wang, L., Singhal, A., and Jajodia, S., "Measuring network security using dynamic bayesian network," Proceedings of the 4th ACM workshop on Quality of protection, 2008, pp. 23–30.

[11] Raya, M., Manshaei, M. H., Félegyházi, M., and Hubaux, J.-P., "Revocation games in ephemeral networks," Proceedings of the 15th ACM Conference on Computer and Communications Security, 2008, pp. 199–210.

[12] Raya, M., Papadimitratos, P., Aad, I., Jungels, D., and Hubaux, J.-P., "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 8, 2007, pp. 1557–1568.

[13] Bilogrevic, I., Manshaei, M. H., Raya, M., and Hubaux, J.-P., "Optimal revocations in ephemeral networks: A game-theoretic framework," Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt), 2010, pp. 21–30.

[14] Behfarnia, A. and Eslami, A., "Error correction coding meets cyber-physical systems: Message-passing analysis of self-healing interdependent networks," *IEEE Transactions on Communications*, Vol. 65, No. 7, 2017, pp. 2753–2768.

[15] Behfarnia, A. and Eslami, A., "Message passing for analysis and resilient design of self-healing interdependent cyber-physical networks," 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, 2016, pp. 1–6.

[16] Behfarnia, A. and Eslami, A., "Dynamics and steady-state behavior of self-healing cyber-physical networks in light of cyber-node delays," IEEE Globecom Workshops (GC Wkshps), Washington DC, 2016, pp. 1–6.

[17] Behfarnia, A. and Eslami, A., "Risk assessment of autonomous vehicles using Bayesian defense graphs," 88th Vehicular Technology Conference (VTC2018-Fall), Chicago, IL, 2016, pp. 1–5.

[18] Behfarnia, A. and Eslami, A., "Local voting games for misbehavior detection in VANETs in presence of uncertainty," 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2019, pp. 480–486.

[19] Behfarnia, A. and Eslami, A., "Misbehavior detection in ephemeral networks: A local voting game in presence of uncertainty," *IEEE Access*, Vol. 7, 2019, pp. 184629–184642.

[20] Buldyrev, S. V., Shere, N. W., and Cwilich, G. A., "Interdependent networks with identical degrees of mutually dependent nodes," *Physical Review E*, Vol. 83, No. 1, 2011, pp. 016112.

[21] Parshani, R., Buldyrev, S. V., and Havlin, S., "Interdependent networks: Reducing the coupling strength leads to a change from a first to second order percolation transition," *Physical review letters*, Vol. 105, No. 4, 2010, pp. 048701.

[22] Schneider, C. M., Yazdani, N., Araújo, N. A., Havlin, S., and Herrmann, H. J., "Towards designing robust coupled networks," *Scientific reports*, Vol. 3, 2013.

[23] Shao, J., Buldyrev, S. V., Havlin, S., and Stanley, H. E., "Cascade of failures in coupled network systems with multiple support-dependent relations," *arXiv preprint arXiv:1011.0234*, 2010.

[24] Gao, J., Buldyrev, S. V., Havlin, S., and Stanley, H. E., "Robustness of a network formed by n interdependent networks with a one-to-one correspondence of dependent nodes," *Physical Review E*, Vol. 85, No. 6, 2012, pp. 066134.

[25] Yağan, O., Qian, D., Zhang, J., and Cochran, D., "Optimal allocation of interconnecting links in cyber-physical systems: Interdependence, cascading failures, and robustness," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 9, 2012, pp. 1708–1720.

[26] Huang, Z., Wang, C., Stojmenovic, M., and Nayak, A., "Balancing system survivability and cost of smart grid via modeling cascading failures," *IEEE Transactions on Emerging Topics in Computing*, Vol. 1, No. 1, 2013, pp. 45–56.

[27] Huang, Z., Wang, C., Nayak, A., and Stojmenovic, I., "Small cluster in cyber physical systems: Network topology, interdependence and cascading failures," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 26, No. 8, 2015, pp. 2340–2351.

[28] Shahrivar, E. M., Pirani, M., and Sundaram, S., "Robustness and algebraic connectivity of random interdependent networks," *IFAC-PapersOnLine*, Vol. 48, No. 22, 2015, pp. 252–257.

[29] Pandurangan, G. and Trehan, A., "Xheal: localized self-healing using expanders," Proceedings of the 30th annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, ACM, 2011, pp. 301–310.

[30] Gallos, L. K. and Fefferman, N. H., "Simple and efficient self-healing strategy for damaged complex networks," *Phys. Rev. E*, Vol. 92, Nov 2015, pp. 052806.

[31] Quattrociocchi, W., Caldarelli, G., and Scala, A., "Self-healing networks: redundancy and structure," *PloS one*, Vol. 9, No. 2, 2014, pp. e87986.

[32] Liu, C., Li, D., Fu, B., Yang, S., Wang, Y., and Lu, G., "Modeling of self-healing against cascading overload failures in complex networks," *EPL (Europhysics Letters)*, Vol. 107, No. 6, 2014, pp. 68003.

[33] Drakopoulos, K., Ozdaglar, A., and Tsitsiklis, J. N., "A lower bound on the performance of dynamic curing policies for epidemics on graphs," 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 3560–3567.

[34] Li, Y.-F., Sansavini, G., and Zio, E., "Non-dominated sorting binary differential evolution for the multi-objective optimization of cascading failures protection in complex networks," *Reliability Engineering & System Safety*, Vol. 111, 2013, pp. 195–205.

[35] Stippinger, M. and Kertész, J., "Enhancing resilience of interdependent networks by healing," *Physica A: Statistical Mechanics and its Applications*, Vol. 416, 2014, pp. 481–487.

[36] Majdandzic, A., Braunstein, L. A., Curme, C., Vodenska, I., Levy-Carciente, S., Stanley, H. E., and Havlin, S., "Multiple tipping points and optimal repairing in interacting networks," *Nature communications*, Vol. 7, 2016.

[37] Kschischang, F. R., Frey, B. J., and Loeliger, H. A., "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, Vol. 47, No. 2, Feb 2001, pp. 498–519. doi:10.1109/18.910572.

[38] Shokrollahi, A., "LDPC codes: An introduction," *Digital Fountain, Inc., Tech. Rep*, 2003, pp. 2.

[39] Jin, H., Khandekar, A., and McEliece, R., "Irregular repeat-accumulate codes," Proc. 2nd Int. Symp. Turbo codes and related topics, Citeseer, 2000, pp. 1–8.

[40] Richardson, T. J. and Urbanke, R. L., "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, Vol. 47, No. 2, 2001, pp. 599–618.

[41] Divsalar, D., Dolinar, S., and Pollara, F., "Iterative turbo decoder analysis based on density evolution," *IEEE journal on selected areas in communications*, Vol. 19, No. 5, 2001, pp. 891–907.

[42] Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E., and Havlin, S., "Catastrophic cascade of failures in interdependent networks," *Nature*, Vol. 7291, No. 464, April 2010, pp. 1025–1028.

[43] Barabási, A.-L. and Albert, R., "Emergence of scaling in random networks," *science*, Vol. 286, No. 5439, 1999, pp. 509–512.

[44] Bollobás, B., *Random graphs*, Springer, 1998.

[45] Dobson, I., Carreras, B., and Newman, D., "A branching process approximation to cascading load-dependent system failure," 37th Annual Hawaii International Conference on System Sciences, Hawaii, USA, January 2004, pp. 1–10.

[46] Jo, K., Kim, J., Kim, D., Jang, C., and Sunwoo, M., "Development of autonomous car – Part I: Distributed system architecture and development process," *IEEE Transactions on Industrial Electronics*, Vol. 61, No. 12, 2014, pp. 7131–7140.

[47] Jo, K., Kim, J., Kim, D., Jang, C., and Sunwoo, M., "Development of autonomous car – Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 8, 2015, pp. 5119–5132.

[48] Gerla, M., Lee, E.-K., Pau, G., and Lee, U., "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014, pp. 241–246.

[49] Dresner, K. and Stone, P., "Mitigating catastrophic failure at intersections of autonomous vehicles," Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1393–1396.

[50] Barabási, A.-L., Albert, R., and Jeong, H., "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A: statistical mechanics and its applications*, Vol. 281, No. 1, 2000, pp. 69–77.

[51] Liu, Y. and Man, H., "Network vulnerability assessment using Bayesian networks," Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005, Vol. 5812, 2005, pp. 61–72.

[52] Kordy, B., Pièvre-Cambacédès, L., and Schweitzer, P., "DAG-based attack and defense modeling: Do not miss the forest for the attack trees," *Computer science review*, Vol. 13, 2014, pp. 1–38.

[53] Kordy, B., Pièvre-Cambacédès, L., and Schweitzer, P., "DAG-based attack and defense modeling: Don't miss the forest for the attack trees," *Computer Science Review*, Vol. 13, 2014, pp. 1–38.

[54] Frigault, M. and Wang, L., "Measuring network security using bayesian network-based attack graphs," Proceedings of the 3rd IEEE International Workshop on Security, Trust, and Privacy for Software Applications (STPSA 08), 2008, pp. 698–703.

[55] Sun, X., Dai, J., Liu, P., Singhal, A., and Yen, J., "Using Bayesian networks for probabilistic identification of zero-day attack paths," *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 10, 2018, pp. 2506–2521.

[56] Xie, P., Li, J. H., Ou, X., Liu, P., and Levy, R., "Using Bayesian networks for cyber security analysis," IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2010, pp. 211–220.

[57] Poolsappasit, N., Dewri, R., and Ray, I., "Dynamic security risk management using bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 1, 2012, pp. 61–74.

[58] Hein, G. W., Kneissi, F., Avila-Rodriguez, J.-A., and Wallner, S., "Authenticating GNSS Proofs Against Spoofs," *Inside GNSS*, September/October 2007, pp. 71–78.

[59] Humphreys, T. E., "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 2, APRIL 2013, pp. 1073–1090.

[60] Preston, S., "GPS multipath detection and mitigation timing bias techniques," *Auburn University Thesis*, 2015.

[61] Moser, D., Leu, P., Lenders, V., Ranganathan, A., Ricciato, F., and Capkun, S., "Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures," ACM Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, New York City, NY, October 2016, pp. 375–386.

[62] Schäfer, M., Lenders, V., and Schmitt, J., "Secure track verification," IEEE Symposium on Security and Privacy (SP), San jose, CA, 2015, pp. 199–213.

[63] Magiera, J. and Katulski, R., "Detection and mitigation of GPS spoofing based on antenna array processing," *Journal of Applied Research and Technology*, Vol. 13, No. 1, 2015, pp. 45–57.

[64] Nielsen, J., Broumandan, A., and Lachapelle, G., "GNSS spoofing detection for single antenna handheld receivers," *Navigation*, Vol. 58, No. 4, 2011, pp. 335–344.

[65] Daneshmand, S., Broumandan, A., Sokhandan, N., and Lachapelle, G., "GNSS multipath mitigation with a moving antenna array," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 1, 2013, pp. 693–698.

[66] Wesson, K. D., Evans, B. L., and Humphreys, T. E., "A combined symmetric difference and power monitoring GNSS anti-spoofing technique," IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013, pp. 217–220.

[67] Jafarnia-Jahromi, A., Broumandan, A., Nielsen, J., and Lachapelle, G., "Pre-despreading authenticity verification for GPS L1 C/A signals," *Navigation*, Vol. 61, No. 1, 2014, pp. 1–11.

[68] Papadimitratos, P. and Jovanovic, A., "GNSS-based positioning: Attacks and countermeasures," IEEE Military Communications Conference (MILCOM), 2008, pp. 1–7.

[69] Khanafseh, S., Roshan, N., Langel, S., Chan, F.-C., Joerger, M., and Pervan, B., "GPS spoofing detection using RAIM with INS coupling," IEEE/ION Position, Location and Navigation Symposium-PLANS, 2014, pp. 1232–1239.

[70] Li, L., Yang, M., Guo, L., Wang, C., and Wang, B., "Hierarchical neighborhood based precise localization for intelligent vehicles in urban environments," *IEEE Transactions on Intelligent Vehicles*, Vol. 1, No. 3, Sept 2016, pp. 220–229.

[71] Zhao, Y., "Applying time-differenced carrier phase in non-differential GPS/IMU tightly coupled navigation systems to improve the positioning performance," *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 2, 2017, pp. 992–1003.

[72] Brown, R. G., "A baseline GPS RAIM scheme and a note on the equivalence of three RAIM methods," *Navigation*, Vol. 39, No. 3, 1992, pp. 301–316.

[73] Salós, D., Martineau, A., Macabiau, C., Bonhoure, B., and Kubrak, D., "Receiver autonomous integrity monitoring of GNSS signals for electronic toll collection," *IEEE transactions on intelligent transportation systems*, Vol. 15, No. 1, 2014, pp. 94–103.

[74] Blanch, J., Walker, T., Enge, P., Lee, Y., Pervan, B., Rippl, M., Spletter, A., and Kropp, V., "Baseline advanced RAIM user algorithm and possible improvements," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 1, 2015, pp. 713–732.

[75] Hewitson, S. and Wang, J., "Extended receiver autonomous integrity monitoring (e raim) for gnss/ins integration," *Journal of Surveying Engineering*, Vol. 136, No. 1, 2010, pp. 13–22.

[76] Roysdon, P. F. and Farrell, J. A., "GPS-INS outlier detection & elimination using a sliding window filter," American Control Conference (ACC), 2017, IEEE, 2017, pp. 1244–1249.

[77] Humphreys, T. E., Ledvina, B. M., Psiaki, M. L., O'Hanlon, B. W., and Kintner Jr, P. M., "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," Proceedings of the ION GNSS International Technical Meeting of the Satellite Division, Vol. 55, 2008, p. 56.

[78] Wesson, K. D., Shepard, D. P., Bhatti, J. A., and Humphreys, T. E., "An evaluation of the vestigial signal defense for civil GPS anti-spoofing," Proceedings of the ION GNSS Meeting, 2011, pp. 1–11.

[79] Macher, G., Sporer, H., Berlach, R., Armengaud, E., and Kreiner, C., "SAHARA: a security-aware hazard and risk analysis method," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, pp. 621–624.

[80] Islam, M. M., Lautenbach, A., Sandberg, C., and Olovsson, T., "A risk assessment framework for automotive embedded systems," Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, 2016, pp. 3–14.

[81] Henniger, O., Apvrille, L., Fuchs, A., Roudier, Y., Ruddle, A., and Weyl, B., "Security requirements for automotive on-board networks," 9th International Conference on Intelligent Transport Systems Telecommunications,(ITST), 2009, pp. 641–646.

[82] Jafarnia-Jahromi, A., Broumandan, A., Nielsen, J., and Lachapelle, G., "GPS vulnerability to spoofing threats and a review of antispoofing techniques," *International Journal of Navigation and Observation*, Vol. 2012, No. 127072, July 2012, pp. 1–16.

[83] Warner, J. S. and Johnston, R. G., "GPS spoofing countermeasures," *Homeland Security Journal*, Vol. 25, No. 2, 2003, pp. 19–27.

[84] Psiaki, M. L. and Humphreys, T. E., "GNSS spoofing and detection," *Proceedings of the IEEE*, Vol. 104, No. 6, 2016, pp. 1258–1270.

[85] Frigault, M., Wang, L., Singhal, A., and Jajodia, S., "Measuring network security using dynamic bayesian network," Proceedings of the 4th ACM workshop on Quality of protection, 2008, pp. 23–30.

[86] Manshaei, M. H., Zhu, Q., Alpcan, T., Bacşar, T., and Hubaux, J.-P., "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, Vol. 45, No. 3, 2013, pp. 25.

[87] Rahim, A., Kong, X., Xia, F., Ning, Z., Ullah, N., Wang, J., and Das, S. K., "Vehicular social networks: A survey," *Pervasive and Mobile Computing*, Vol. 43, 2018, pp. 96–113.

[88] Yin, L., Guo, Y., Li, F., Sun, Y., Qian, J., and Vasilakos, A., "A game-theoretic approach to advertisement dissemination in ephemeral networks," *World Wide Web*, Vol. 21, No. 2, 2018, pp. 241–260.

[89] Qiu, T., Chen, B., Sangaiah, A. K., Ma, J., and Huang, R., "A survey of mobile social networks: Applications, social characteristics, and challenges," *IEEE Systems Journal*, Vol. 12, No. 4, 2018, pp. 3932–3947.

[90] Akrida, E. C., Gasieniec, L., Mertzios, G. B., and Spirakis, P. G., "Ephemeral networks with random availability of links: The case of fast networks," *Journal of Parallel and Distributed Computing*, Vol. 87, 2016, pp. 109–120.

[91] van der Heijden, R., Dietzel, S., and Kargl, F., "Misbehavior detection in vehicular ad-hoc networks," *1st GI/ITG KuVS Fachgesprach Inter-Vehicle Communication. University of Innsbruck*, 2013, pp. 23–25.

[92] Shivshankar, S. and Jamalipour, A., "An evolutionary game theory-based approach to cooperation in VANETs under different network conditions," *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 5, 2015.

[93] Hasrouny, H., Samhat, A. E., Bassil, C., and Laouiti, A., "Misbehavior detection and efficient revocation within VANET," *Journal of Information Security and Applications*, Vol. 46, 2019, pp. 193–209.

[94] van der Heijden, R. W., Dietzel, S., Leinmüller, T., and Kargl, F., "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys & Tutorials*, 2018.

[95] Loukas, G., Karapistoli, E., Panaousis, E., Sarigiannidis, P., Bezemskij, A., and Vuong, T., "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles," *Ad Hoc Networks*, Vol. 84, 2019, pp. 124–147.

[96] Liu, Y., Comaniciu, C., and Man, H., "Modelling misbehaviour in ad hoc networks: A game theoretic approach for intrusion detection," *International Journal of Security and Networks*, Vol. 1, No. 3-4, 2006, pp. 243–254.

[97] Manshaei, M. H., Zhu, Q., Alpcan, T., Bacşar, T., and Hubaux, J.-P., "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, Vol. 45, No. 3, 2013, pp. 25.

[98] Hendrikx, F., Bubendorfer, K., and Chard, R., "Reputation systems: A survey and taxonomy," *Journal of Parallel and Distributed Computing*, Vol. 75, 2015, pp. 184–197.

[99] Liu, B., Chiang, J. T., and Hu, Y.-C., "Limits on revocation in VANETs," 8th International Conference on Applied Cryptography and Network Security, 2010, pp. 38–52.

[100] Abass, A. A. A., Mandayam, N. B., and Gajic, Z., "An evolutionary game model for threat revocation in ephemeral networks," 51st Annual Conference on Information Sciences and Systems (CISS), 2017, pp. 1–5.

[101] Lekha, S. I. and Kathiroli, R., "Trust based certificate revocation of malicious nodes in MANET," International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 1185–1189.

[102] Kim, S., "Effective certificate revocation scheme based on weighted voting game approach," *IET Information Security*, Vol. 10, No. 4, 2016, pp. 180–187.

[103] Masdari, M., "Towards secure localized certificate revocation in mobile Ad-hoc networks," *IETE Technical Review*, Vol. 34, No. 5, 2017, pp. 561–571.

[104] Arshad, M., Ullah, Z., Ahmad, N., Khalid, M., Criuckshank, H., and Cao, Y., "A survey of local/cooperative-based malicious information detection techniques in VANETs," *EURASIP Journal on Wireless Communications and Networking*, Vol. 2018, No. 1, 2018, pp. 62.

[105] Diakonikolas, I. and Pavlou, C., "On the complexity of the inverse semi-value problem for weighted voting games," Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 1869–1876.

[106] Subba, B., Biswas, S., and Karmakar, S., "Intrusion detection in mobile Ad-hoc networks: Bayesian game formulation," *Engineering Science and Technology, an International Journal*, Vol. 19, No. 2, 2016, pp. 782–799.

[107] Subba, B., Biswas, S., and Karmakar, S., "A game theory based multi layered intrusion detection framework for VANET," *Future Generation Computer Systems*, Vol. 82, 2018, pp. 12–28.

[108] Kerrache, C. A., Lakas, A., Lagraa, N., and Barka, E., "UAV-assisted technique for the detection of malicious and selfish nodes in VANETs," *Vehicular Communications*, Vol. 11, 2018, pp. 1–11.

[109] Silva, C. R., Moraes, R. O., Lelis, L. H., and Gal, K., "Strategy generation for multi-unit real-time games via voting," *IEEE Transactions on Games*, 2018.

[110] Esmaeilyfard, R., Hendessi, F., Manshaei, M. H., and Grossklags, J., "A game-theoretic model for users' participation in ephemeral social vehicular networks," *International Journal of Communication Systems*, Vol. 32, No. 12, 2019, pp. 3998.

[111] Hof, F., Kern, W., Kurz, S., Pashkovich, K., and Paulusma, D., "Simple games versus weighted voting games: bounding the critical threshold value," *Available at SSRN 3270445*, 2018.

[112] Ferdowsi, A., Challita, U., Saad, W., and Mandayam, N. B., "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 307–312.

[113] Yu, B., Xu, C.-Z., and Xiao, B., "Detecting sybil attacks in VANETs," *Journal of Parallel and Distributed Computing*, Vol. 73, No. 6, 2013, pp. 746–756.

[114] Fudenberg, D. and Tirole, J., *Game Theory*, MIT Press, 1991.

[115] Sanguesa, J. A., Naranjo, F., Torres-Sanz, V., Fogue, M., Garrido, P., and Martinez, F. J., "On the study of vehicle density in intelligent transportation systems," *Mobile Information Systems*, Vol. 2016, 2016.

[116] contributors, W., "Markov decision process — Wikipedia, The Free Encyclopedia," https://en.wikipedia.org/w/index.php?title=Markov_decision_process&oldid=889290458, 2019.

[117] Xiao, L., Xie, C., Chen, T., Dai, H., and Poor, H. V., "A mobile offloading game against smart attacks," *IEEE Access*, Vol. 4, 2016, pp. 2281–2291.

[118] Xiao, L., Wan, X., Dai, C., Du, X., Chen, X., and Guizani, M., "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, Vol. 25, No. 3, 2018, pp. 116–122.

[119] Li, J., Gao, H., Lv, T., and Lu, Y., "Deep reinforcement learning based computation offloading and resource allocation for MEC," 2018 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2018, pp. 1–6.

[120] Jaszkiewicz, A., "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, 2002, pp. 402–412.