Detecting Malicious Nodes Using Game Theory and Reinforcement Learning in Software-Defined Networks

Amir Javadpour, Forough Ja'fari, Tarik Taleb, and Chafika Benzaïd

Abstract

Mafia, or Werewolf, is a strategic game where two teams compete to eliminate each other's players through deception and hidden roles. The game dynamics and role interactions share notable similarities with adversarial behaviors in network security, making it a valuable framework for modeling cyber threats, particularly botnet detection. In this paper, we introduce a novel game-theoretic approach to botnet detection, leveraging the strategic deception dynamics of the Mafia game to model adversarial behavior in cybersecurity. We present a mathematical model for Mafia games, formulating winning strategies for different roles using linear relations and reinforcement learning techniques. Furthermore, we establish a direct mapping between Mafia game roles and network security components, illustrating how botnet attack patterns align with hidden-role game mechanics. Our proposed detection strategies are applied to real-world network attack scenarios, demonstrating their effectiveness in mitigating botnet threats. We evaluate the model using applicable security metrics and compare the results with existing detection methodologies to validate the approach. Our findings indicate that the suggested strategies improve detection accuracy by 12% over conventional methods. Additionally, we conduct network emulations using Mininet, simulating Mirai botnet infections. The results show that the true positive and true negative detection rates for a network modeled by the Mafia game framework reach 71% and 91%, respectively. These insights provide a foundation for integrating deception-based modeling into modern intrusion detection systems, enhancing network resilience against adaptive cyber threats.

Index Terms

Mafia game, Reinforcement learning, Network security, Mirai botnet, Software-defined network (SDN), Intrusion detection, Honeypot, Anti-malware, Moving target defense (MTD).

I. Introduction

THE diminution in the functionality of a network after a cyber attack is undeniable. Several tools and mechanisms, such as Intrusion Detection Systems (IDSs), honeypots and deceptions, and Moving Target Defense (MTD), are currently proposed for securing the network [1, 2, 3, 4]. However, a general model that considers all of these mechanisms in a network is not still presented. This general model helps us in evaluating the performance of the whole security solution. The general network security problem is modeled with the Mafia game for the first time in this paper.

The current researches on the Mafia game have tried to balance the game or to give a strategy for some of the roles. However, they have some limitations, such as the problem of generality and the weakness of suggested strategies. The current mathematical models cannot fully cover the Mafia game, considering its different roles and parameters. A mathematical model makes it easier for the researchers to analyze the game balance, and also provides a way of applying the game concepts to other fields in order to solve their related problems. Moreover, the existing suggested strategies are not general and they only consider some specific parameters. To overcome the aforementioned limitations, we have proposed a mathematical model for representing the Mafia game, as well as suggesting more general strategies for the different roles of this game. Additionally, the similarities between a Mafia game and a real network under a botnet attack, which is briefly illustrated in Figure 1, motivate us to first define the Mafia model as a general game theory model, and then to model the real networks with that. This modeling is beneficial because we can use the known strategies of the Mafia game as security strategies in a network.

The key contributions of this paper are as follows:

- Defining the Mafia model as a new game model that could be used in game theory concepts.
- Modeling general network security aspects as a Mafia game in order to increase the detection rate of the malicious nodes.
- Suggesting the detection strategies that are efficient in detecting the Mafias in a game and the malicious nodes in a network.
- Proposing the algorithms that can apply the suggested strategies on the game model, using two different techniques.

Amir Javadpour is with ICTFICIAL Oy, Espoo, Finland. He was with the Faculty of Information Technology and Electrical Engineering, University of Oulu when this research work was initiated (e-mail: a.javadpour87@gmail.com).

Forough Ja'fari is with the Department of Computer Engineering, Yazd University, Safaeih, Yazd, 100190, Iran. (e-mail: azadeh.mth@gmail.com).

Tarik Taleb is with the Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum, Germany (e-mail: tarik.taleb@rub.de).

Chafika Benzaïd is with the faculty of Information Technology and Electrical Engineering, University of Oulu (e-mail: chafika.benzaid@oulu.fi).

Corresponding author: Amir Javadpour

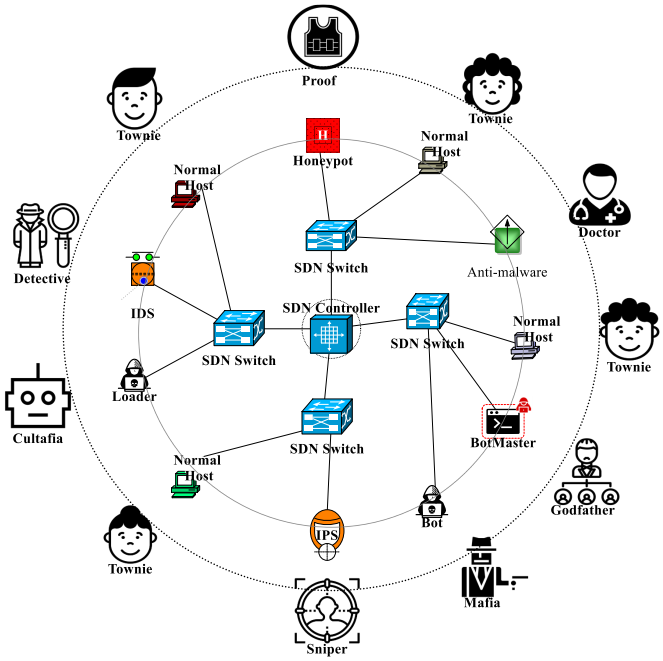


Fig. 1. The mapping between the Mafia game roles and the network components [5]

- Suggesting new metrics based on previously introduced common metrics for evaluating the performance of Mafia game players.

In this study, we propose a new framework that models network security challenges using game-theoretic principles inspired by the game of Mafia. This approach provides a structured way to analyze adversarial interactions in a network, where malicious actors engage in strategic behavior to evade detection. However, the applicability of this model to the real world, especially in terms of its advantages over existing cybersecurity frameworks, needs further elucidation. A key aspect of this research is to demonstrate how the proposed game-theoretic model enhances network security beyond conventional methods by improving detection accuracy and decision-making processes.

To validate our model, we conduct extensive experiments using network simulations in Mininet, where we simulate various attack scenarios, including botnet infiltration and hostile manipulation. While the effectiveness of our approach is demonstrated through promising detection rates, a more detailed description of the experimental setup is essential for reproducibility. Key parameters such as network topology, attack settings, and reinforcement learning model specifications are important in evaluating the model performance. Furthermore, the interpretation of the experimental results requires a deeper analysis, especially in terms of false positive and false negative rates, which are crucial for assessing the reliability of the model. By addressing these aspects, we aim to provide a comprehensive assessment of the applicability of our model and ensure its relevance to real-world network security challenges.

The remaining of this paper is structured as follows. section II provides a detailed background description of both Mafia games and network security mechanisms. section III reviews the current researches on the Mafia game, and its limitations. In section IV we have explained the mathematical game model of Mafia, considering the players, rules, belief functions, and winning strategies. The algorithms for applying the suggested strategies are also mentioned in this section. The way of modeling a network and its security aspects are described in section V, and section VI provides the evaluation results of the suggested strategies and the network modeling. Finally, our plan for future work and the conclusion of this paper is presented in section VII and section IX.

II. Background

In this section, the background of the Mafia game and also network security tools are presented.

A. Mafia game

Mafia is a role-based party game, invented by Dimitry Davidoff in 1986, and it is also called the Werewolf game. This game includes two groups of players: Mafias and Townies. The goal of each team is to eliminate the players of the other team from the game. The Townies do not know the role of each player, so they try to find clues about the Mafias. On the other hand, the Mafias know each other, but they attempt to lure the Townies by pretending to be a Townie. The game has a single Narrator who controls the game flow and is aware of all the roles and actions. The game consists of day and night phases one after the

other, and the Narrator announces the shifts between these phases. In a day phase, the players talk to each other, and at the end of that phase, the Narrator performs a voting process and the players vote to choose the one who must exit the game. The player receiving the most votes is removed from the game. It must be noted that the number of votes must be more than a certain threshold. After a day phase, we have a night phase. In the night phase, all the players, except the Mafias, close their eyes. The Mafias shot one of the players and that player will exit the game the next day. The game finishes in one of these two conditions: (1) when all the Mafias exit the game, which leads to the winning of Townies, or (2) when the number of Mafias is equal to the number of Townies, which leads to the winning of the Mafias. In addition, to be a Mafia or a Townie, each player has a specific role in the game, and some of these roles have specific abilities at night. For example, a Detective, as a Townie player, can select a player at night and ask the Narrator if that player is a Mafia, and the Narrator will answer with a thumbs up or down. As another example, a Godfather, as a Mafia player, is the leader of the Mafia team and cannot be detected by the detective [6].

Here are some of the main roles in a simple Mafia game along with their characteristics and abilities:

- **Godfather:** The Godfather is a powerful leader in the Mafia team, and there is only a single Godfather in this team. The final decision of selecting a target to be shot in the night phase is made by the Godfather. Moreover, when the Godfather is the target of the Detective, the Narrator will answer with a thumbs down. In other words, the Detective cannot detect the Godfather, and is not sure whether or not that player is a Townie of a Godfather.
- **Cultafia:** The Cultafia is one of the Mafia team players, who can select one of the Townies during the night phase, and invite it to join the Mafia team. If the target player does not have a specific role (i.e. a Normal Townie), it will be converted to a Mafia. This process is called negotiation. The Cultafia must monitor the players during the day phase to find the players who probably do not have a night ability.
- **Mafia (Normal Mafia):** The (Normal) Mafia is in the Mafia team, and does not have a specific ability during the night phase. The (Normal) Mafias just communicate with the other Mafia team players during the night phase. Most of the (Normal) Mafias pay attention to the Godfather's targets during the day phase. The Godfather signals them to agree on a single target to vote.
- **Detective:** The Detective is in the Townie team and can detect the Mafia team players, except the Godfather. In the night phase, the Detective targets one of the players, and the Narrator responds with a thumbs up or down. Thumbs up mean the target player is a Mafia or Cultafia, and thumbs down are for other players. This process is called requesting. The Detective must be careful in selecting its target, by finding a player who is probable to be in the Mafia team.
- **Doctor:** The Doctor is one of the Townie team players, who can save one of the players in the night phase. If that player is shot by the Mafia team, it will remain in the game. This player must pay attention to the game flow to select the players who are probable to be shot in the night phase.
- **Proof:** The Proof is a Townie player and does not perform a task during the night phase. However, if it is shot by the Mafia team in the night phase or selected by the votes in the day phase, it will remain in the game. If the Proof is selected by the player's votes for being removed from the game, the Narrator will tell all the players that it is Proof. Therefore, the Mafia team must be careful in voting. The players who have voted the Proof are very suspicious to be in the Mafia team.
- **Townie (Normal Townie):** The (Normal) Townie is in the Townie team, and does not have a specific ability during the night phase. However, their power is in the votes they made. They can find each other by their votes or speeches, and they can form a group to vote for a single player who is probably a Mafia.

It is worth noting that the mentioned game scenario is one out of hundreds of scenarios that are considered for this game. For example, in some Mafia game scenarios, there is also another team of players, called the Masons, the players which can communicate during the night phase. Or as another example, in some scenarios, the Proof has a shield, by which it could remain in the game only after a specific number of shots. Moreover, the number of shots, inquests, and similar activities, or even the names of different roles may change in different scenarios. However, the mentioned scenario is the most simple and common one, and hereafter, this particular scenario will be referred to with the general name of the Mafia game.

Now, we give an example of Mafia game procedures. Figure 2 illustrates the different phases and what happens in each phase. Ten players are participating in this game: a Godfather, a Cultafia, a Mafia, a Detective, a Doctor, a Proof, and four Townies. On the first day, the players talk to each other, and at the end of that day, they decide to vote. However, since none of the players gets enough votes to be removed, all the players remain in the game. The next night, the Detective inquests about player 9. Because in its opinion, player 9 was suspicious the previous day. However, the Narrator responds with a thumbs down. Because the target player is a Townie. Since Doctor thinks player 7 is on its side, according to their similar vote on the previous day, player 7 is saved by Doctor. However, this save is useless, because the Mafia team has shot player 5. As a result, player 5 is removed from the game. On the second day, the Mafia team decides to push player 4 out. So Mafia and Godfather vote for player 4. But Cultafia tries to separate itself from its team to not be detected. Some of the other Townies are also deceived by the Mafia team to vote for player 4. However, when the voting process is finished, the Narrator announces that player 4 is Proof, and no one will be removed. In the next night, since the Detective is suspicious about the players who tried to push Proof out, detects one of them, like player 3. As player 3 is Mafia, the Narrator responds with a thumbs up. On

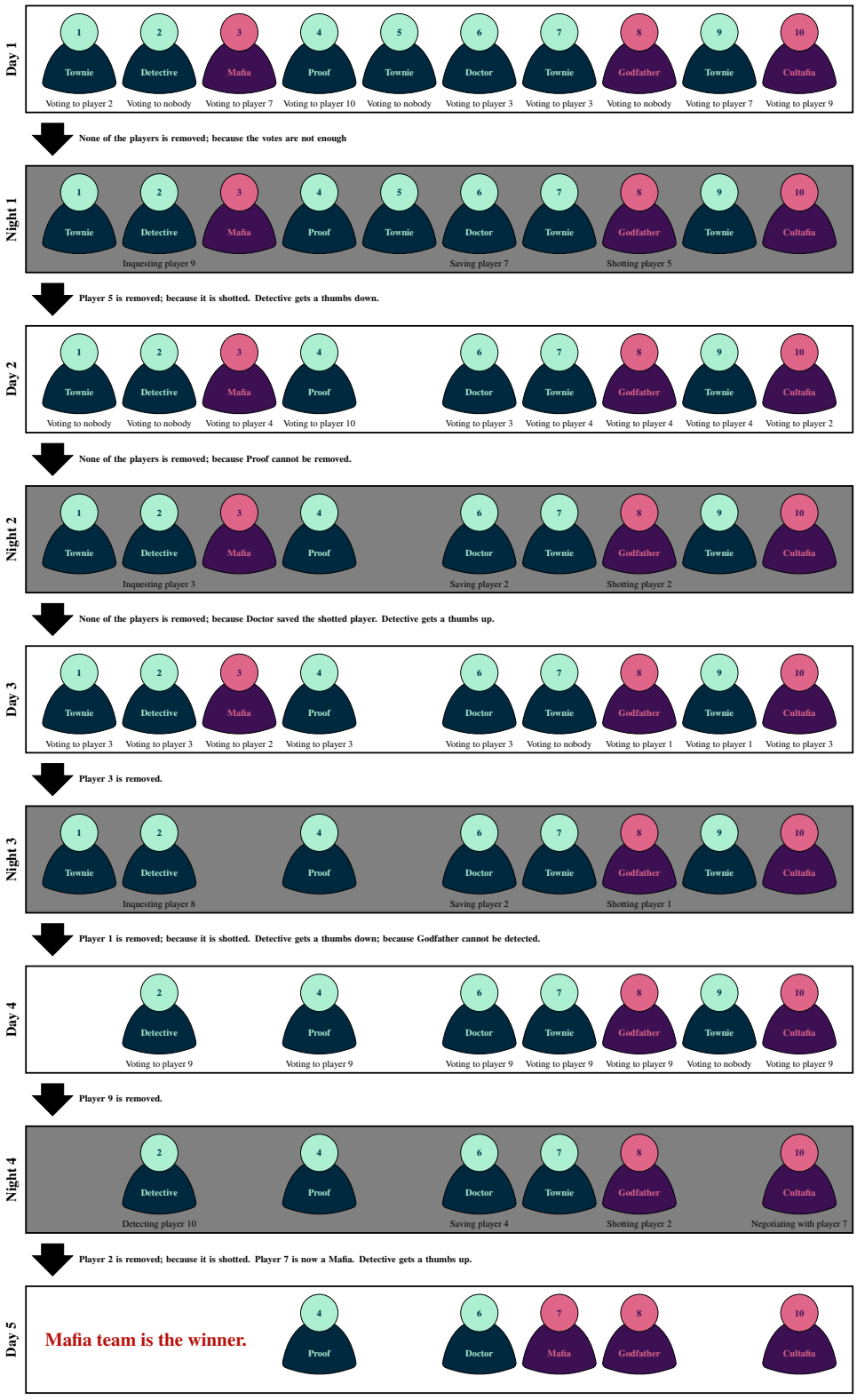


Fig. 2. A sample Mafia game with 10 players

the other hand, since player 2 did not vote for anyone in the past days, Doctor and Godfather guess that player 2 is Detective. As a result, the Mafia team shots player 2, and Doctor also saves it. So, no one will be removed. Due to the suspicion of player 3 and its thumbs-up situation, most of the Townies vote for player 3 on the third day. Moreover, Cultafia also votes for its teammate to not be detected. At the end of the third day, player 3 is removed from the game. The next night, Detective asks for detecting player 8. Because this player did not vote for the removed Mafia (i.e. player 3), and its votes were on the

side of that Mafia. However, since player 8 is Godfather, the Narrator responds with a thumbs down. Since Doctor has found the Detective during the game flow, player 2 is again saved by Doctor. The Mafia team does not shoot player 2. Because on the last night that player was saved, and on this night it is probable to be saved again. They shot player 1, and this player is removed. On the fourth day, the Townie team decide to vote for player 9. Because the first suspicious player was player 9. But Detective gets a thumbs down. So they think player 9 is a Mafia. Therefore, player 9 is removed at the end of the fourth day. In the next night, the Detective inquests player 10, and the Narrator responds with a thumbs up. The doctor changes its target and saves player 4 this time. However, Godfather shots player 2. Moreover, Cultafia negotiates with player 7. Because that player is probable to have no abilities in night phases. So, at the end of this night, player 2 is removed, and player 7 is converted to a Mafia. On the fifth day, the number of players in the Mafia and Townie teams are equal, and hence, the Mafia team is the winner, and the game finishes.

B. Network security

The computer networks may face different security threats, by which they lose their functionalities and the ability to provide their services. One of the most powerful ways of launching an attack against the network is to create a botnet and propagate it among the network nodes. Most of the botnets consist of a command and control server, a botnet script loader, and several bots [7]. The adversary controls its bot army using the command and control server. This component sends the signals to the bots or loaders in order to manage their activities. For example, the command and control server sends the address of a specific host in the network to all the bots and commands them to launch an attack against it. When flooding traffic is sent from distributed malicious nodes toward a target to deny its availability, a Distributed Denial of Service (DDoS) attack is formed. The loader in a botnet is responsible for infecting the hosts to convert them into a bot. The loader loads the botnet script on the victim host, and after executing this script, the victim becomes a bot. The bots also follow the commands of the adversary to perform distributed tasks, such as scanning the network or launching an attack.

On the other hand, there are also some security tools in a computer network. Different mechanisms have been proposed to detect and prevent botnet attacks. These mechanisms have been divided into the following three categories [8, 9]:

- Mechanisms that act during the formation phase of the botnet and prevent vulnerable hosts from becoming bots or out-of-function. Anti-malware programs and Data Execution Prevention (DEP) mechanisms, which can prevent the execution of malicious scripts or malware on the hosts, are some of the members of this category. Also, Identity Access Management (IAM) techniques that block the access of external nodes to the local hosts, may be put in this same category.
- Mechanisms that detect the final attack. This category includes the IDSs, which use techniques such as anomaly detection and statistical analysis of network traffic to detect malicious behavior and malicious nodes.
- Mechanisms that try to prevent the occurrence of the final attack. This category includes Intrusion Prevention Systems (IPSs), honeypots, and MTD approaches. An IPS is a security tool that monitors the network and tries to prevent harmful activities in different ways, such as blocking the malicious source or reporting them to the network administrator. A honeypot is a deceptive trap in the network that pretends to be a normal host. When the malicious nodes communicate with a honeypot, they are closely monitored, and then detected. MTD is a security mechanism that invalidates the adversary's information by changing the attack space (e.g. by changing the address of the host).

Note that different techniques are used by these security tools to make their final decision. For example, an IDS receives a dataset of malicious patterns, and based on these patterns, decides whether or not the current traffic is malicious. This decision can be made by machine learning models. An example of a machine learning model is reinforcement learning. In a reinforcement learning model, the problem is passed to a learning agent, and the agent explores different solutions to find the best one. On the other hand, some the security tools use mathematical relations in order to find malicious nodes. For example, they create a linear relation of the parameters they have, and then based on the value of this equation, they decide whether or not the current traffic is sorted as legal.

Moreover, the evolution of Software-Defined Networks (SDNs) has improved the performance of these tools. Because in SDN, the traffic flow is controlled and a general view of the network is available. Hence, some actions, such as blocking the malicious nodes or extracting the attack patterns, can be performed more effectively. The management of network traffic is the responsibility of the controller in SDNs.

III. Literature Review

Recent advances in machine learning and behavioral analysis have introduced new perspectives in botnet detection. Supervised and unsupervised learning techniques have been employed to analyze network traffic, identify anomalies, and distinguish between legitimate and malicious activities[10, 11, 12]. However, these methods often face limitations in terms of generalization, requiring large labeled datasets and extensive retraining to remain effective against evolving threats. Studies such as [13, 14, 15, 16] have explored deep learning approaches for botnet detection, focusing on feature extraction and real-time network flow classification. Additionally, Moving Target Defense (MTD) mechanisms have been proposed to dynamically alter network configurations, reducing the predictability of attack surfaces and making it more difficult for adversaries to establish long-term control over compromised devices [13, 14, 15]. Despite these advancements, botnet operators continue to employ sophisticated

TABLE I
A SUMMARY OF THE MAIN RELATED RESEARCHES

Research Group	Reference	Key Parameter	Main Achievement	Proposing a Model	Applying to Other Fields
Game balancing	[23]	Event probabilities	Player's enjoyment	✗	✗
	[24]			✗	✗
	[25]			✓	✗
	[26]	User experience		✗	✗
	[27]	Game refinement		✗	✗
	[28]			✗	✗
Strategy devising	[29]	Psychological behavior	Helping the players	✗	✗
	[30]			✗	✗
	[31]			✗	✗
	[32]			✗	✗
	[33]			✗	✗
	[34]			✗	✗
	[35]		Training an agent	✗	✗
	[36]			✗	✗
	[37]			✗	✓
	[38]			✗	✗
	[39]			✗	✗
	[40]	Logical facts	Helping the players	✓	✗
	[41]			✗	✗
	[42]			✗	✗
	Our work			✓	✓
	[43]		Training an agent	✗	✗
	[44]			✓	✗
	[45]			✗	✗
	[46]			✓	✗
	[47]			✗	✗
	[48]			✗	✗
	[49]			✗	✗
	[50]			✗	✗
	[51]			✗	✗
	[52]			✗	✗
	[53]			✗	✗

evasion tactics, such as fast-flux networks, domain generation algorithms (DGA), and peer-to-peer (P2P) architectures, making it increasingly difficult to track and neutralize command-and-control (C2) servers. These tactics resemble the strategic deception found in the Mafia game, where adversaries operate concealed while coordinating attacks indirectly [17, 18]. In the Mafia framework, the Godfather, representing the botnet control server, orchestrates attacks while remaining undetected, whereas the Mafia members, corresponding to infected nodes, execute malicious actions based on concealed strategies. Drawing from this analogy, our approach integrates game-theoretic principles into network security, allowing for the simulation of adversarial interactions, strategy optimization, and improved detection of hidden threats[5]. Several existing works have applied game theory to cybersecurity, demonstrating its effectiveness in modeling strategic behaviors. For example, Stackelberg game models have been employed to simulate attacker-defender dynamics in network intrusion scenarios, while Bayesian games have been used to assess threat probabilities and optimize resource allocation for defense mechanisms [19, 20, 21, 22]. However, these approaches often assume fully rational players and perfect information, limiting their applicability in real-world adversarial settings where attackers exhibit deceptive and adaptive behavior. The Mafia game framework, in contrast, explicitly incorporates uncertainty and deception, providing a more realistic model for botnet detection. By expanding upon these insights, our work aims to bridge the gap between traditional network security methods and game-theoretic approaches, demonstrating the potential of the Mafia framework as an effective tool for analyzing, predicting, and mitigating botnet threats.

The researches that have been conducted on the Mafia game, which are summarized in Table I, are categorized into different groups. The researches in the first group focused on the game parameters to provide a good gameplay experience for the players. Braverman et al. [23] and Yao [24] have examined the different number of players of each Mafia and Townie team in order to find the optimal number of players, by which the winning probability of both teams is nearly the same at the beginning of the game. Migdał [25] has proposed a mathematical model based on the game events (e.g. the winning of Mafia/Townie) probabilities to reach a similar goal. Xu et al. [26] have studied the impact of players' dominance adjustment on their enjoyment. Xiong et al. [27] and Ri et al. [28] have studied the measure of Mafia game refinement to balance the game.

The researchers in the second group have suggested some strategies for one of these purposes: (1) helping the players win on their side, or (2) proposing an automated agent that can play the Mafia game like a human. Some of these works are based on human psychological behavior, such as their body language or the phrases they are using. The others are based on the logical facts that are obtained from the game, such as the player's votes.

Zhou and Sung [29] and Demyanov et al. [30] have studied the problem of finding the players who lie during the game using their used phrases and facial expressions, respectively. Katagami et al. [31] have investigated the impact of nonverbal information, such as nose touching or arm folding, on the team's winning. Girlea et al. [32] have investigated the power

of psycholinguistic features in detecting the deceptive players. Nagayama et al. [33] have investigated the impact of Mafia member's cooperation, such as whispering to each other, on their winning. Tellols [34] have studied the impact of talkativeness on the performance of the Mafia game players.

Katagami et al. [35] have proposed an agent that learns the nonverbal information of the real-world players in a Mafia game and then acts like them. Hirata et al. [36] have also proposed an artificial intelligence model that learns human behavior, especially communication skills, and can play the Mafia game automatically. Kano et al. [37] have used the conversations between the players during the Mafia game for evaluating the dialog systems. Ibraheem et al. [38] have used neural networks to detect the liars in a Mafia game based on their use of language. Khan and Aranha [39] have proposed an agent that uses ensemble learning, that predicts the optimal strategy by combining the results of multiple machine learning models. The models in this work are trained based on the positive or negative sentences that are used by the other players. These features help the agent in predicting who will vote for it.

Gîrlea et al. [40] have modeled the player's belief based on their conversations. An exact strategy is not mentioned in this work. However, its proposed model helps the players find their strategy. Halim [41] has proposed some strategies for the Townies, Mafias, Detective, and Doctor, based on the probabilities of different events. For example, it is mentioned that the Doctor must not reveal its role, because it may be the night shot target. However, the suggested strategies are not adequate for considering them as a winning strategy. Bi and Tanaka [42] have suggested some strategies for the Mafia players, especially for the Mafia members to make themselves similar to the Townies. In this work, it is assumed that the Detective reveals its role in the first phase of the game. The general strategies related to our work, which we call as **"Previous Strategy 1"**, are as follows:

- The Detective must randomly select its targets.
- The Doctor must select the targets that are previously saved from the shots.
- The Godfather must persist on the previously failed shots.

A platform for playing the Mafia game has been proposed by Toriumi et al. [54], in which the players can play with an agent that utilizes artificial intelligence. The main focus of this work is to make the agent change its behavior based on its role and propose a communication protocol, and not to find a winning strategy. Wang et al. [55] have also proposed a robotic agent for playing the Mafia game, and its main focus is on the hardware and communication protocol.

Hatori et al. [43] have suggested a strategy of detecting the liars in a Mafia game where one or more players have revealed their role. Lin et al. [44] have also suggested the strategies of guessing the player's roles in a game that the players reveal their role or a fake role. This work has proposed a mathematical model for the game. Lin et al. [45] have performed a similar work using supervised classifier learning. Nide and Takata [46] have used the belief-desire–intention model for modeling the Mafia game, by which the players can trust or not trust the players who have revealed their roles.

Azaria et al. [47] have proposed a machine learning agent for detecting deceptive players based on their discussion features, such as the number of accusing sentences, in an online Mafia game. Nakamura et al. [48] have proposed an agent that can automatically play the Mafia game based on the psychological model. This model indicates the belief of each player about the other players' roles by assigning them a score based on the probabilities. The suggested strategies for this work, which are related to the scope of our work are as follows:

- If the Detective is in the game, the Doctor must save the player who is probably the Detective. Otherwise, the Doctor must select a random player among the players who are probably a Townie.
- The Detective must inquire about the player who is probably a Mafia.
- If the Detective or the Doctor are in the game, the Godfather must shoot the player who is probably the Detective or the Doctor. Otherwise, the target is randomly selected.

Kondoh et al. [49] have used long short-term memory learning in order to design an efficient agent for the Mafia game. The features that are used for training the agent are the conversation between the players and their votes for all the other players. Hagiwara et al. [50] have also considered the conversations and the votes in order to train a Mafia agent that utilizes reinforcement learning. Wang and Kaneko [51] have proposed an agent that uses deep reinforcement learning for playing the Mafia game. We called the strategies of this agent as **"Previous Strategy 2"**. The features, which are considered in this agent and also are related to the scope of our paper, are as follows:

- The number of each player's votes to each of the other players.
- The state of being active or inactive for all the players.
- The previous night-phase actions of The Godfather and the Detective for their own belief.

Eger and Martens [52] have proposed a mechanism, by which an agent can figure out the best time for changing its plans. Chang [53] has proposed a simulator for analyzing the Mafia game balance, and also suggested some strategies. This simulator assigns a credibility weight to each of the players, by which some actions are automatically performed. Its suggested strategy for the Mafia team is to first shoot the Detective, if its role is revealed, and otherwise shot the player with the highest credibility weight. We call this strategy as **"Previous Strategy 3"**. The credibility weight of a player is increased every time that player votes for a revealed Mafia, or the revealed Mafia vote for it.

The reviewed researches are limited regarding the following aspects:

- The existing mathematical models are not general enough for covering all the facts of the game. Moreover, they do not have considered some specific roles in the game, such as the Proof or Cultafia.
- The current suggested strategies do not consider some events that happened in the games with extra roles, such as the Proof and Cultafia.
- The existing reinforcement learning model cannot be trained with n players, and then used for m players, where $n \neq m$. They can be used only for the same game parameters that were trained.
- None of the reviewed researches have used the concept of the Mafia game to model other fields' problems, especially network security. To the best of our knowledge, only a single research has used the Mafia game for evaluating dialog systems [37]. However, it does not have presented the required descriptions of this usage and evaluations for the proposed method.
- All the reviewed researches lack defining an evaluation metric for the performance of different roles in the Mafia game. They have only considered the winning results. But the winning results are not suitable for separately evaluating different roles.

Due to the mentioned limitations, we have proposed a general mathematical model for representing a Mafia game and suggested strategies for each of the players with extra abilities. We have limited our work to logical facts other than the players' conversations because detectingay be complicated, while the gamets based on the acti such as the voted night shots, are more reliable.

IV. Proposed Mafia Model

One of our goals in this paper is to model a network with a Mafia game and then detect the malicious nodes of the network using the detection strategies of this game. To model a network with the Mafia game, it is first required to define the Mafia game model. Hence, in this section, we define the game model and its features, such as the player's belief and the strategies they can apply.

Four main specifications must be defined for each game model: the game players along with their actions and roles, the game rules and their parameters, the belief function of each player, and the game strategies that lead to the winning of a team. In this section, we will describe these specifications to define our Mafia model. In general, a Mafia game can be represented as $\mathcal{M} = \{\mathcal{R}, \mathcal{P}, \mathcal{P}', \mathcal{A}, \mathcal{A}', \mathcal{B}\}$, where $\mathcal{R}$ is the set of game rule parameters. $\mathcal{P}$ and $\mathcal{P}'$ are the player's roles at the beginning of each day and night, respectively. $\mathcal{A}$ and $\mathcal{A}'$ are the history of the player's actions during the game day and night phases, respectively. $\mathcal{B}$ is the set of players' belief parameters. Each of these features is described in detail at the remainder of this section, and the notations, the functions, and the conditions are summarized in Table II, Table III, and Table IV, respectively.

A. Game players

The players in the proposed Mafia model are a Godfather (GO), a Cultafia (CU), multiple Mafias (MA), a Detective (DE), a Doctor (DO), a Proof (PR), and multiple Townies (TO). The number of Mafias and Townies depends on the total number of players. But the point is that in the initial state of the game, the number of Townie team players must be greater than the number of Mafia team players. We can say that $\mathcal{P}$ and $\mathcal{P}'$ are the matrices that represent the player's roles at the beginning of each day and night, respectively. The element in their i^{th} row and the j^{th} column is shown by $p_{i,j}$ and $p'_{i,j}$, respectively. $p_{i,j}$ and $p'_{i,j}$ are the role of the j^{th} player at the beginning of the i^{th} day and night, respectively. Moreover, we have $p_{i,j}, p'_{i,j} \in \mathbb{P}$, where $\mathbb{P} = \{1 : GO, 2 : CU, 3 : MA, 4 : DE, 5 : DO, 6 : PR, 7 : TO, 8 : PR'\}$. Note that PR' is the Proof that was the target of votes, but its role was announced by the Narrator. The values of $\mathcal{P}$ and $\mathcal{P}'$ for the sample game shown in Figure 2 are presented in Equation 1 and Equation 2.

$$\mathcal{P}_{\text{sample}} = \begin{bmatrix} 7 & 4 & 3 & 6 & 7 & 5 & 7 & 1 & 7 & 2 \\ 7 & 4 & 3 & 6 & \phi & 5 & 7 & 1 & 7 & 2 \\ 7 & 4 & 3 & 8 & \phi & 5 & 7 & 1 & 7 & 2 \\ \phi & 4 & \phi & 8 & \phi & 5 & 3 & 1 & 7 & 2 \\ \phi & \phi & \phi & 8 & \phi & 5 & 3 & 1 & \phi & 2 \end{bmatrix} \tag{1}$$

$$\mathcal{P}'_{\text{sample}} = \begin{bmatrix} 7 & 4 & 3 & 6 & 7 & 5 & 7 & 1 & 7 & 2 \\ 7 & 4 & 3 & 8 & \phi & 5 & 7 & 1 & 7 & 2 \\ 7 & 4 & \phi & 8 & \phi & 5 & 7 & 1 & 7 & 2 \\ \phi & \phi & \phi & 8 & \phi & 5 & 3 & 1 & \phi & 2 \end{bmatrix} \tag{2}$$

Each role has some specific abilities, some of which are common between different roles. All the players can vote for their desired target at the end of each day. The Detective can inquest a player in the night phase. The Doctor is also capable of saving a player in a night phase. The Cultafia can negotiate with a player to convert it to a Mafia. The Godfather is the one who can shoot a player in the night phases.

We represent all the actions that are performed during the day and night phase as two matrices, called $\mathcal{A}$ and $\mathcal{A}'$, respectively. These matrices contain P columns, and the element in their i^{th} row and j^{th} column are shown as $a_{i,j}$ and $a'_{i,j}$, respectively.

TABLE II
THE LIST OF NOTATIONS USED IN THE PROPOSED MAFIA MODEL

Notation	Description
$\mathcal{M}$	The Mafia game
$\mathcal{R}$	The set of game rule parameters
μ	The votes threshold for removing a player in percents
α	The shots threshold for the Godfather
α'	The shots threshold for the Godfather in each night
β	The negotiations threshold for the Cultafia
β'	The negotiations threshold for the Cultafia in each night
λ	The inquesting threshold for the Detective
λ'	The inquesting threshold for the Detective in each night
δ	The saves threshold for the Doctor
δ'	The saves threshold for the Doctor in each night
P	The total number of players
$\mathbb{P}$	The set of defined roles
$\mathcal{P}$	The players and their roles in the day phases
$p_{i,j}$	The j^{th} player's roles in the i^{th} day
$\mathcal{P}'$	The players and their roles in the night phases
$p'_{i,j}$	The j^{th} player's roles in the i^{th} night
$\mathcal{A}$	The history of the players' actions during the day phases
$a_{i,j}$	The j^{th} player's action in the i^{th} day
$\mathcal{A}'$	The history of the players' actions during the night phases
$a'_{i,j}$	The j^{th} player's action in the i^{th} night
$\mathcal{B}$	The set of players' belief parameters
$\mathcal{B}_{GO}$	The set of Godfather's belief parameters
$\mathcal{B}_{CU}$	The set of Cultafia's belief parameters
$\mathcal{B}_{DE}$	The set of Detective's belief parameters
$\mathcal{B}_{DO}$	The set of Doctor's belief parameters
Θ	Godfather's belief function
Ψ	Cultafia's belief function
Ω	Detective's belief function
Γ	Doctor's belief function

TABLE III
THE LIST OF DEFINED FUNCTIONS IN THE PROPOSED MAFIA MODEL

Function	Description	Equation
$ro_j^k(i)$	A function that returns the set of player numbers with the role of i in the j^{th} day or k^{th} night.	Equation 5
$ro_m(i)$	A function that returns the set of players in the Mafia team at the i^{th} day or night.	Equation 13
$ro_t(i)$	A function that returns the set of players in the Townie team at the i^{th} day or night.	Equation 14
$tv_j(i)$	A function that returns the number of votes for the i^{th} player at the end of the j^{th} day.	Equation 6
$ac_j(i)$	A binary function that returns 1 if the i^{th} player is in the game at the end of the j^{th} day before voting.	Equation 7
$in_j(i)$	A binary function that returns 1 if the i^{th} player is not in the game at the beginning of the j^{th} night.	Equation 21
$ma_j(i)$	A binary function that returns 1 if the i^{th} player is in the Mafia team at the beginning of the j^{th} night.	Equation 22
$bsu(i,j)$	A binary function that returns 1 if the i^{th} player is survived by the shot in the j^{th} night.	Equation 23
$su_j(i)$	A function that returns the number of time the i^{th} player survived after being shot till the beginning of the j^{th} night.	Equation 24
$bvo(i,j)$	A binary function that returns 1 if the i^{th} player vote for some player at the end of the j^{th} day.	Equation 25
$vo_j(i)$	A function that returns the number of times the i^{th} player takes roles in voting till the beginning of the j^{th} night.	Equation 26
$bco(i,j)$	A binary function that returns 1 if the i^{th} player vote for one of the Mafia players at the end of the j^{th} day.	Equation 27
$co_j(i)$	A function that returns the number of i^{th} player's votes for one of the Mafia members till the beginning of the j^{th} night.	Equation 28
$fa_j(i)$	A binary function that returns 1 if the i^{th} player is the target of a failed negotiation till the beginning of the j^{th} night.	Equation 29
$bex(i,j,k)$	A binary function that returns 1 if the i^{th} player has voted for the k^{th} player at the end of the j^{th} day.	Equation 30
$ex_j(i)$	A function that returns the number of the i^{th} player's votes for the removed Townies till the beginning of the j^{th} night..	Equation 31
$bpr(i,j)$	A binary function that returns 1 if the i^{th} player has voted the Proof at the end of the j^{th} day.	Equation 32
$pr_j(i)$	A function that returns the number of the i^{th} player's votes for a revealed Proof till the beginning of the j^{th} night.	Equation 33
$bpn(i,j)$	A binary function that returns 1 if the Detective gets a thumbs up for the i^{th} player in the j^{th} night.	Equation 34
$po_j(i)$	A binary function that returns 1 if the Detective gets a thumbs up for the i^{th} player till the beginning of the j^{th} night.	Equation 35
$ne_j(i)$	A binary function that returns 1 if the Detective gets a thumbs down for the i^{th} player till the beginning of the j^{th} night.	Equation 35
$bsi(i,j)$	A binary function that returns 1 if the Doctor's and the i^{th} player's votes are equal at the end of the j^{th} day.	Equation 37
$si_j(i)$	A function that returns the common votes between the Doctor and the i^{th} player till the beginning of the j^{th} night.	Equation 38

$a_{i,j}$ and $a'_{i,j}$ are the target of the j^{th} player action in the i^{th} day and night, respectively. It is clear that $a_{i,j}, a'_{i,j} \in \mathbb{N}_P$, where $\mathbb{N}_P$ is the set of natural numbers from 1 to P. For the sample game shown in Figure 2, the day and night actions are shown

TABLE IV
THE LIST OF DEFINED CONDITIONS IN THE PROPOSED MAFIA MODEL RULES

Condition	Description	Equation
$C_{\text{voted}}(i,j)$	The condition of removing the j^{th} player at the end of the i^{th} day	Equation 8
$C_{\text{shot}}(i,j)$	The condition of removing the j^{th} player at the end of the i^{th} night	Equation 9
$C_{\text{changed}}(i,j)$	The condition of changing the j^{th} player to a Mafia at the i^{th} night	Equation 10
$C_{\text{up}}(i,j)$	The condition of a thumbs up for the j^{th} player at the i^{th} night	Equation 11
$C_{\text{down}}(i,j)$	The condition of a thumbs down for the j^{th} player at the i^{th} night	Equation 12
$C_{\text{mafia}}(i)$	The condition of Mafia's winning at the i^{th} day or night	Equation 15
$C_{\text{townie}}(i)$	The condition of Townie's winning at the i^{th} day or night	Equation 16

in Equation 3 and Equation 4, respectively.

$$\mathcal{A}_{\text{sample}} = \begin{bmatrix} 2 & \phi & 7 & 10 & \phi & 3 & 3 & \phi & 7 & 9 \\ \phi & \phi & 4 & 10 & \phi & 3 & 4 & 4 & 4 & 2 \\ 3 & 3 & 2 & 3 & \phi & 3 & \phi & 1 & 1 & 3 \\ \phi & 9 & \phi & 9 & \phi & 9 & 9 & 9 & \phi & 9 \end{bmatrix} \tag{3}$$

$$\mathcal{A}'_{\text{sample}} = \begin{bmatrix} \phi & 9 & \phi & \phi & \phi & 7 & \phi & 5 & \phi & \phi \\ \phi & 3 & \phi & \phi & \phi & 2 & \phi & 2 & \phi & \phi \\ \phi & 8 & \phi & \phi & \phi & 2 & \phi & 2 & \phi & \phi \\ \phi & 10 & \phi & \phi & \phi & 1 & \phi & 2 & \phi & 7 \end{bmatrix} \tag{4}$$

We also define a function, called $ro(i)$, that returns the set of active players of role i. For example, $ro(1)$ in the sample game is $\{8\}$. Because 1 represents the Godfather role (GO), and Godfather is the eighth player. The j^{th} member of this set is shown by $ro(i,j)$ and its value in the k^{th} day or night is shown by $ro_k()$ or $ro^k()$, respectively. This function is calculated by Equation 5.

$$\begin{aligned} ro_k(i) &= \{l \in \mathbb{N}_P : p_{k,l} = i\} \\ ro^k(i) &= \{l \in \mathbb{N}_P : p'_{k,l} = i\} \end{aligned} \tag{5}$$

B. Game rules

In this section, the Mafia game rules are mentioned. It is worth noting that in different Mafia game scenarios, these rules are different. We define the adequate rules for our model.

The players are only permitted to select one of the valid actions defined for their role. $\mathcal{R}$ contains the numerical thresholds for these actions, where $\mathcal{R} = \{\mu, \alpha, \alpha', \beta, \beta', \lambda, \lambda', \delta, \delta'\}$. In the voting process, a player leaves the game when its number of votes reaches μ percent of the active players in the game. The Godfather can shoot up to α players, and the Cultafia can negotiate with up to β players in the whole game. But none of them can shoot or negotiate with more than α' and β' players in each phase. The Detective can detect up to λ players, and the Doctor can save up to δ players in the whole game. But none of them can detect or save more than λ' and δ' players in each phase, respectively. It must be noted that in our game mode, each player can vote up to one player at the end of the day. In the sample game shown in Figure 2, we have $\mathcal{R} = \{35, 7, 1, 1, 1, 4, 2, 5, 2\}$. It means that if the number of votes for a player reaches 35% of the active players, it will be removed. The Mafia team can shoot 7 players, the Cultafia can negotiate with only one player, the Detective can only detect 4 players, and the Doctor can save 5 players. It is worth noting that in the sample game, the Doctor and the Detective can have two targets each night, but they prefer to only use one of their chances.

At the end of each day phase, when the voting process is over, the player with the most number of votes is removed considering the μ threshold. If the number of votes are equal for two or more players, none of them will be removed. $C_{\text{voted}}(i,j)$ is the condition, by which the i^{th} player is removed from the game at the end of the j^{th} day. For presenting this condition, we define a function, called $tv_j(i)$, that returns the total votes for the i^{th} player at the end of the j^{th} day. It can be calculated as Equation 6.

$$tv_j(i) = |\{k \in \mathbb{N}_P : a_{j,k} = i\}| \tag{6}$$

Moreover, we define a binary function, called $ac_j(i)$, that returns 1 if the i^{th} player is in the game at the end of the j^{th} day, before applying the voting results. This function is calculated based on Equation 7.

$$ac_j(i) = \begin{cases} 1, & \text{If } p_{j,i} \neq \phi \\ 0, & \text{Otherwise} \end{cases} \tag{7}$$

Now the condition, in which the j^{th} player is removed because of the votes at the end of the i^{th} day, is presented in Equation 8.

$$C_{\text{voted}}(i,j) = \begin{cases} 0(False), & \text{If } \exists k \in \mathbb{N}_P : tv_i(j) \leq tv_i(k) \\ 0(False), & \text{Else If } tv_i(j) < \mu \times \sum_{k=1}^{P} ac_i(k) \\ 1(True), & \text{Otherwise} \end{cases} \tag{8}$$

When the Godfather shots a player, that player will be removed at the beginning of the next day, only if it is not the Proof and it is not saved by the Doctor. This condition is presented in Equation 9, where $C_{\text{shot}}(i,j)$ is the condition, in which the j^{th} player is successfully shot at the i^{th} night.

$$C_{\text{shot}}(i,j) = \begin{cases} 0(False), & \text{If } p'_{i,j} = ro(6) \\ 0(False), & \text{Else If } a'_{i,ro(5,0)} = j \\ 0(False), & \text{Else If } a'_{i,ro(1,0)} \neq j \\ 1(True), & \text{Otherwise} \end{cases} \tag{9}$$

The Cultafia's negotiation is successful, only if the target player does not have a specific role. So, we can define the condition, in which the j^{th} player is successfully negotiated at the i^{th} night, is presented in Equation 10.

$$C_{\text{changed}}(i,j) = \begin{cases} 0(False), & \text{If } a'_{i,ro(2,0)} \neq j \\ 1(True), & \text{Else If } p'_{i,j} = 7 \\ 0(False), & \text{Otherwise} \end{cases} \tag{10}$$

The Detective can inquest a player at night, and the Narrator will respond with a thumbs up only if that player is in the Mafia team and is not the Godfather. In other situations, a thumbs down is the response. These conditions are shown in Equation 11 and Equation 12.

$$C_{\text{up}}(i,j) = \begin{cases} 0(False), & \text{If } a'_{i,ro(4,0)} \neq j \\ 1(True), & \text{Else If } p'_{i,j} \in \{2,3\} \\ 0(False), & \text{Otherwise} \end{cases} \tag{11}$$

$$C_{\text{down}}(i,j) = \begin{cases} 0(False), & \text{If } a'_{i,ro(4,0)} \neq j \\ 0(False), & \text{Else If } p'_{i,j} \in \{2,3\} \\ 1(True), & \text{Otherwise} \end{cases} \tag{12}$$

Now, we define the conditions in which the game is over. The criteria for winning and losing the game for the Townie and Mafia teams are defined to be:

- If the number of Mafia members is greater than or equal to the number of Townie members, the game will end in favor of the Mafia team. This condition is shown in Equation 15, where $ro_m(i)$ and $ro_t(i)$ are the set of Mafia and Townie players at the i^{th} day or night, which are calculated based on Equation 13 and Equation 14, respectively. If the value of $C_{\text{mafia}}(i)$ is True in the i^{th} day or night, that day or night is the last phase of the game and Mafia wins.

$$ro_m(i) = ro_i^i(1) \cup ro_i^i(2) \cup ro_i^i(3) \tag{13}$$

$$ro_t(i) = ro(4) \cup ro_i^i(5) \cup ro_i^i(6) \cup ro_i^i(7) \cup ro_i^i(8) \tag{14}$$

$$C_{\text{mafia}}(i) = \begin{cases} 1(True), & \text{If } |ro_m(i)| \geq |ro_t(i)| \\ 0(False), & \text{Otherwise} \end{cases} \tag{15}$$

- If the Godfather is removed from the game, the game will end in favor of the Townie team. This condition is presented in Equation 16. If the value of $C_{\text{townie}}(i)$ is True at the end of the i^{th} night or day, the Townie team is the winner.

$$C_{\text{townie}}(i) = \begin{cases} 1(True), & \text{If } |ro_i^i(1)| = 0 \\ 0(False), & \text{Otherwise} \end{cases} \tag{16}$$

C. Players belief

The significant feature of the Mafia games is the uncertainty. The Townies are not sure which of the other players are in the Townie team, and on the other hand, the Mafia team is not sure about selecting their target for the votes, shots, or negotiations. However, a belief function can be defined for each of the players. This function contains the essential facts that a player has collected during the game, and it is updated during the game phases. The output of a belief function is an ordered set of the players, which is sorted by the priority of the players to be the target. For example, if the output of the Detective's belief function is $\{3, 2, 4, 5, 1\}$, it means that the third player has a higher probability to be a Mafia. So, it is preferred to be detected by the Detective. And if the detective wants to detect two players in that night, the third and the second players must be its choices.

In this paper, we are not focusing on the day phase actions, which is in our case the voting process. Hence, we just define the belief function for the players who have a night-phase ability:

- **Godfather:** The night-phase action of a Godfather is to shot an appropriate player. The Godfather defines four different degrees for each of the players, which are $in()$, $ma()$, $su()$, and $vo()$. $in(i)$ ($inactive(i)$) is a binary degree, which is 1 if the i^{th} player is not in the game. $ma(i)$ ($mafia(i)$) is a binary degree, which is 1 if the i^{th} player is in the Mafia team. $su(i)$ ($survive(i)$) is the number of times the i^{th} player survived after being shot. $vo(i)$ ($vote(i)$) is the total number of times the i^{th} player takes role in voting. These four degrees can help the Godfather select its targets for shot. The prototype of the Godfather's belief function is shown in Equation 17.

$$\text{Godfather's belief} = \Theta(\bigcup_{i \in \mathbb{N}_P} \{in(i), ma(i), su(i), vo(i)\}) \tag{17}$$

- **Cultafia:** The Cultafia is the negotiator of the Mafia team. The defined degrees by the Cultafia are $in()$, $ma()$, $co()$ and $fa()$. $in()$ and $ma()$ are the same degrees defined by the Godfather. $co(i)$ ($correct(i)$) is the number of the i^{th} player's votes for one of the Mafia team players. $fa(i)$ ($failed(i)$) is a binary degree that is 1 if the i^{th} player was negotiated before, but the negotiation failed. If the Cultafia pays attention to these four degrees, its negotiations may be more probable to be successful. This belief function is shown in Equation 18.

$$\text{Cultafia's belief} = \Psi(\bigcup_{i \in \mathbb{N}_P} \{in(i), ma(i), co(i), fa(i)\}) \tag{18}$$

- **Detective:** Detecting the Mafia players is the responsibility of the Detective. The Detective defines five different degrees, called $in()$, $ex()$, $pr()$, $po()$, and $ne()$, for each of the players at the end of each day. $in()$ is the same degree defined before. $ex(i)$ ($exit(i)$) is the number of removed players that were in the Townie team and the i^{th} player has voted them. $pr(i)$ ($proof(i)$) is the number of the i^{th} player's votes for the Proof, only if the Proof roles is announced to all the players by the Narrator. $po(i)$ ($positive(i)$) and $ne(i)$ ($negative(i)$) are binary degrees that are 1 if the i^{th} player caused a thumbs up and down in the Detective's previous inquestings, respectively. Considering these five degrees, the Detective can detect the Mafia team with a higher performance. The Detective's belief function is shown in Equation 19.

$$\text{Detective's belief} = \Omega(\bigcup_{i \in \mathbb{N}_P} \{in(i), ex(i), pr(i), po(i), ne(i)\}) \tag{19}$$

- **Doctor:** The Doctor tries to save the players who are probable to be shot by the Mafia team. Four degrees are defined by the Doctor, which are $in()$, $ex()$, $pr()$, and $si()$. The $in()$, $ex()$, and $pr()$ degrees are the same as the Detective's defined degrees. $si(i)$ ($side(i)$) is the number of common votes between the i^{th} player and the Doctor. According to these four degrees, the Doctor may save the players who are more probable to be shot than the other players. This belief function is shown in Equation 20.

$$\text{Doctor's belief} = \Gamma(\bigcup_{i \in \mathbb{N}_P} \{in(i), ex(i), pr(i), si(i)\}) \tag{20}$$

It must be noted that only the active players are included in the output of the belief functions. A total of eleven different degrees are used in the definition of the belief functions. These degrees construct the set of belief parameters. So, we have $\mathcal{B} = \{\mathcal{B}_{GO}, \mathcal{B}_{CU}, \mathcal{B}_{DE}, \mathcal{B}_{DO}\}$, where $\mathcal{B}_{GO} = \{in(), ma(), su(), vo()\}$, $\mathcal{B}_{CU} = \{in(), ma(), co(), fa()\}$, $\mathcal{B}_{DE} = \{in(), ex(), pr(), po(), ne()\}$, and $\mathcal{B}_{DO} = \{in(), ex(), pr(), si()\}$.

Now we give the equations for calculating the different degrees. The value of $in()$ at the beginning of the j^{th} night is called $in_j(i)$, and it is calculated based on Equation 21.

$$in_j(i) = \begin{cases} 1, & \text{If } p_{j,i} = \phi \\ 0, & \text{Otherwise} \end{cases} \tag{21}$$

$ma(i)$ is 1 if the i^{th} player is in the Mafia team, and otherwise it is 0. Hence, we can calculate the value of $ma(i)$ at the beginning of the j^{th} night ($ma_j(i)$) by Equation 22.

$$ma_j(i) = \begin{cases} 1, & \text{If } p_{j,i} \in \{1,2,3\} \\ 0, & \text{Otherwise} \end{cases} \tag{22}$$

For calculating the $su()$ degree, we first define a binary function, called $bsu(i,j)$, which returns 1 if the i^{th} player is survived by the shot in the j^{th} night. This function can be calculated by Equation 23.

$$bsu(i,j) = \begin{cases} 1, & \text{If } a'_{j,ro^j(1,0)} = i \wedge p'_{j,i} \neq \phi \\ 0, & \text{Otherwise} \end{cases} \tag{23}$$

Now, the value of $su()$ for the i^{th} player at the beginning of the j^{th} night can also be calculated by Equation 24.

$$su_j(i) = \sum_{k=1}^{j} bsu(i,k) \tag{24}$$

For calculating the value of $vo()$, first we define a binary function, $bvo(i,j)$ that returns 1 if the i^{th} player votes for somebody at the end of the j^{th} day. This function is calculated by Equation 25.

$$bvo(i,j) = \begin{cases} 1, & \text{If } a_{j,i} \neq \phi \\ 0, & \text{Otherwise} \end{cases} \tag{25}$$

The value of $vo()$ at the beginning of the j^{th} night ($vo_j()$) can be calculated based on Equation 26.

$$vo_j(i) = \sum_{k=1}^{j} bvo(i,k) \tag{26}$$

Again, for calculating the value of $co()$, we first define a binary function, called $bco(i,j)$ that returns 1 if the i^{th} player votes for one of Mafia team players at the end of the j^{th} day. We can calculate this function with Equation 27.

$$bco(i,j) = \begin{cases} 1, & \text{If } a_{j,i} \in ro^j(1) \cup ro_j(2) \cup ro_j(3) \\ 0, & \text{Otherwise} \end{cases} \tag{27}$$

Considering bco, we can calculate the value of $co()$ at the beginning of the j^{th} day, using Equation 28.

$$co_j(i) = \sum_{k=1}^{j} bco(i,k) \tag{28}$$

$fa_j(i)$ is a binary degree that indicates whether the i^{th} player is the target of a failed negotiation before the j^{th} night. This function is calculated by Equation 29.

$$fa(i,j) = \begin{cases} 1, & \text{If } \exists k < j : a'_{k,ro^k(2,0)} = i \wedge p'_{k,i} \neq 3 \\ 0, & \text{Otherwise} \end{cases} \tag{29}$$

$ex_j(i)$ is the number of Townie team players that are removed from the game and the i^{th} player has voted them before the beginning of the j^{th} night. Before calculating this function, we define a binary function, called $bex(i,j,k)$, that returns 1 if the i^{th} player has voted for the k^{th} player at the end of the j^{th} day, where the k^{th} player is in the Townie team. This function is calculated by Equation 30.

$$bex(i,j,k) = \begin{cases} 1, & \text{If } a_{j,i} = k \wedge ma_j(i) = 0 \\ 0, & \text{Otherwise} \end{cases} \tag{30}$$

Now, the value of $ex()$ is calculate using Equation 31.

$$ex_j(i) = \sum_{k=1}^{P} \sum_{l=1}^{j} bex(i,l,k) \times (1 - in_j(k)) \tag{31}$$

To calculate $pr()$, we first define a binary function, called $bpr(i,j)$, that returns 1 if the i^{th} player has voted the Proof at the end of the j^{th} day. We can calculate this function using Equation 32.

$$bpr(i,j) = \begin{cases} 1, & \text{If } a_{j,i} = ro^j(6,0) \\ 0, & \text{Otherwise} \end{cases} \tag{32}$$

Now the value of $pr_j(i)$, which means the number of the i^{th} players votes at the end of the j^{th} day to the revealed Proof, can be calculated by Equation 33.

$$pr_j(i) = \begin{cases} 0, & \text{If } 8 \notin p'_j \\ \sum_{k=1}^{j} bpr(i,k), & \text{Otherwise} \end{cases} \tag{33}$$

For calculating $po()$ and $ne()$, we define a binary function, called $bpn(i,j)$, that returns 1 if the Detective gets a thumbs up for detecting the i^{th} player at the j^{th} night. Otherwise (i.e. if the Detective does not target the i^{th} player or gets a thumbs down), this function returns 0. We can calculate this function using Equation 34.

$$bpn(i,j) = \begin{cases} 1, & \text{If } a'_{j,ro^j(4,0)} = i \wedge (p_{j,i} = 2 \vee p_{j,i} = 3) \\ 0, & \text{Otherwise} \end{cases} \tag{34}$$

Now, we can calculate $po()$ and $ne()$, using Equation 35 and Equation 36, respectively.

$$po_j(i) = \begin{cases} 1, & \text{If } \exists k < j : bpn(i,k) = 1 \\ 0, & \text{Otherwise} \end{cases} \tag{35}$$

$$ne_j(i) = \begin{cases} 1, & \text{If } \nexists k < j : bpn(i,k) = 1 \wedge \exists k < j : a'_{k,ro^k(4,0)} = i \\ 0, & \text{Otherwise} \end{cases} \tag{36}$$

Again, for calculating the value of $si()$, we define a binary function, called $bsi(i,j)$, that returns 1 if the i^{th} player's vote at the end of the j^{th} day is equal to the Doctors vote. This function is calculated using Equation 37.

$$bsi(i,j) = \begin{cases} 1, & \text{If } a_{j,ro_j(5,0)} = a_{j,i} \\ 0, & \text{Otherwise} \end{cases} \tag{37}$$

Now we can calculate the number of common votes between the Doctor and the i^{th} player until the end of the j^{th} day, using Equation 38.

$$si_j(i) = \sum_{k=1}^{j} bsi(i,j) \tag{38}$$

The mentioned belief functions do not have a specific definition to specify the output. However, in the next part, we will define some algorithms and techniques in order to use these functions toward a winning strategy.

D. Winning strategies

The nature of a Mafia game is uncertain, and there is not a specific strategy which is always correct in detecting the two sides of the game. However, some of the facts in the game can lead to better detection. These facts are used to create the belief functions in the previous part.

The degrees, which are defined by each belief function, give a score to each of the players, and these scores are used to sort them based on their priority in being the target. Some of these degrees can exactly specify, which players must not be in the output set of the belief function. For example, a Cultafia never negotiates with one of its teammates. It is a waste of chance toward the winning of the Mafia team. So, based on the $ma()$ degree, all the Mafia team players are removed from the output of the Cultafia's belief function. However, the exact set and its order are hard to be found.

In this section, we utilize two different techniques in order to find a strategy that is probable to lead one of the teams to win. These techniques are linear relation and reinforcement learning. In the linear relation technique, we consider a linear equation of the degrees to calculate the final score of the players. For example, for the Doctor's belief function, we consider that the score of the i^{th} player is $a.ex(i) + b.pr(i) + c.si(i)$. Then we try to find the best coefficients, which are in our example the values of a, b, and c, based on the Mafia games dataset, that lead to the best result. On the other hand, in the reinforcement learning technique, we do not consider a linear equation between the final score and the degrees. We train a learning model based on the Mafia games dataset, by which we can find a sorted set of the players as the output.

It must be noted that due to the team-based nature of the Mafia game, every action that one of the players does correctly will help their team to win. So all the players try to have the best performance in the game for their team, especially the ones who have a night-phase ability. The winning strategies of these players are as follows:

- **Godfather:** The target players of the night shots must be selected in a way that they are not probable to be saved by the Doctor. Moreover, the probability of being the Proof for the target must be low. The $su()$ degree can help the Godfather in guessing the players who may be saved or who is the Proof. Because there are two reasons that players has been survived.

The first reason is being the Proof, and the second reason is the Doctor saved that player. Both of these reasons specify a bad target for the Godfather. In addition, the players who are not involved too much in the voting processes are good candidates. Because, they probably have a night-phase ability, and they want to be careful in their votes. Hence, the $vo()$ degree can help. It must be noted that the targets of the Godfather must not be selected from the Mafia team members ($ma()$) or inactive players ($in()$). According to these degrees, the winning strategy of the Godfather can be defined by Algorithm 1 and Algorithm 2. In Algorithm 1 a nested loop is considered for finding the best coefficients of $su()$ and

Algorithm 1 The winning strategy of the Godfather using linear relation technique

Require: $\mathcal{B}_{GO}$, the belief parameters of Godfather
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Θ, the ordered set of target candidates
  in, ma, su, $vo \leftarrow \mathcal{B}_{GO}$
  $saves \leftarrow$ the Doctor saves based on $\mathcal{D}$
  $proof \leftarrow$ the Proof based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $best \leftarrow 0$
  $\Theta \leftarrow \{\}$
  for $i \in$ (-20 ... 20) **do**
    for $j \in$ (-20 ... 20) **do**
      $scores \leftarrow \{\}$
      for $k \in (1 \dots P)$ **do**
        $scores \leftarrow scores + \{k, i.su[k] + j.vo[k]\}$
      $scores \leftarrow$ sorted $scores$
      $result \leftarrow 0$
      for $k \in (1 \dots P)$ **do**
        if $scores[k][1] \notin saves \cup \{proof\}$ **then**
          $result \leftarrow result + 1$
        else
          break
      if $result > best$ or $\Theta = \{\}$ **then**
        $best \leftarrow result$
        $\Theta \leftarrow scores[1]$
  for $k \in (1 \dots P)$ **do**
    if $in[k] = 1$ or $ma[k] = 1$ **then**
      $\Theta \leftarrow \Theta$ - $\{k\}$
  return Θ

$vo()$. These coefficients are found according to the number of targets that are not the Proof or saved and they appear at the beginning of the sorted set. When the best coefficients are found, the players are sorted based on the found linear relation scores. Finally, the Mafia members and inactive players are removed from the output set. In Algorithm 2, first a

Algorithm 2 The winning strategy of the Godfather using reinforcement learning technique

Require: $\mathcal{B}_{GO}$, the belief parameters of Godfather
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Θ, the ordered set of target candidates
  in, ma, su, $vo \leftarrow \mathcal{B}_{GO}$
  $saves \leftarrow$ the Doctor saves based on $\mathcal{D}$
  $proof \leftarrow$ the Proof based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $model \leftarrow$ initiate the model
  for $i \in$ (0 ... 1000) **do**
    $scores \leftarrow$ generate the scores using $model$
    $result \leftarrow 0$
    for $k \in (1 \dots P)$ **do**
      if $scores[k][1] \notin saves \cup \{proof\}$ **then**
        $result \leftarrow result + 1$
      else
        break
    update $model$ based on $result$
  $scores \leftarrow$ generate the sorted scores using $model$
  $\Theta \leftarrow scores[1]$
  for $k \in (1 \dots P)$ **do**
    if $in[k] = 1$ or $ma[k] = 1$ **then**
      $\Theta \leftarrow \Theta$ - $\{k\}$
  return Θ

model is trained to learn which scores are more preferable for sorting the players. A set of scores is preferred when its

first players are not the Proof or saved. When the training phase is finished, the players are sorted based on the optimal scores, and then, the Mafia members and inactive players are removed from this set.

- **Cultafia:** To have a successful negotiation, the Cultafia must select a player who does not have a special role. In other words, the target must be a Townie. So, the first thing that must be considered by the Cultafia is not to target one of the Mafia team members, based on $ma()$, or inactive players, based on $in()$. The $co()$ degree is a good metric to reach this goal. Because the Townie team players who have correct targets for the votes may probably have a night-phase ability that gives them extra information. Moreover, the $fa()$ degree also helps. Because the previous players that cause failed negotiation, must not be negotiated again. Hence, the winning strategies for the Cultafia using the linear relation and reinforcement learning techniques are as follows. In Algorithm 3, we have first a loop for finding the sign of $co()$

Algorithm 3 The winning strategy of the Cultafia using linear relation technique

Require: $\mathcal{B}_{CU}$, the belief parameters of Cultafia
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Ψ, the ordered set of target candidates
  in, ma, co, $fa \leftarrow \mathcal{B}_{CU}$
  $roles \leftarrow$ the Detective, Doctor, and Proof based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $best \leftarrow 0$
  $\Psi \leftarrow \{\}$
  for $i \in$ (-2 ... 2) **do**
    $scores \leftarrow \{\}$
    for $k \in (1 \ldots P)$ **do**
      $scores \leftarrow scores + \{k, i.co[k]\}$
    $scores \leftarrow$ sorted $scores$
    $result \leftarrow 0$
    for $k \in (1 \ldots P)$ **do**
      if $scores[k][1] \notin roles$ **then**
        $result \leftarrow result + 1$
      else
        break
    if $result > best$ or $\Psi = \{\}$ **then**
      $best \leftarrow result$
      $\Psi \leftarrow scores[1]$
  for $k \in (1 \ldots P)$ **do**
    if $in[k] = 1$ or $ma[k] = 1$ or $fa[k] = 1$ **then**
      $\Psi \leftarrow \Psi - \{k\}$
  return Ψ

coefficient. It is found according to the sorted set of the player's scores. If the number of players in the first of this set that has no roles are higher, the scores are better. After that, the players are sorted, and then the inactive players, Mafia members, and the previously failed targets are removed from the set. In Algorithm 4, first a model is trained to learn

Algorithm 4 The winning strategy of the Cultafia using a reinforcement learning technique

Require: $\mathcal{B}_{CU}$, the belief parameters of Cultafia
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Ψ, the ordered set of target candidates
  in, ma, co, $fa \leftarrow \mathcal{B}_{CU}$
  $roles \leftarrow$ the Detective, Doctor, and Proof based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $model \leftarrow$ initiate the model
  for $i \in (0 \ldots 1000)$ **do**
    $scores \leftarrow$ generate the scores using $model$
    $result \leftarrow 0$
    for $k \in (1 \ldots P)$ **do**
      if $scores[k][1] \notin roles$ **then**
        $result \leftarrow result + 1$
      else
        break
    update $model$ based on $result$
  $scores \leftarrow$ generate the sorted scores using $model$
  $\Psi \leftarrow scores[1]$
  for $k \in (1 \ldots P)$ **do**
    if $in[k] = 1$ or $ma[k] = 1$ or $fa[k] = 1$ **then**
      $\Psi \leftarrow \Psi - \{k\}$
  return Ψ

which scores are more preferable for sorting the players. A set of scores is preferred when its first players do not have a specific role. When the training phase is finished, the players are sorted based on the optimal scores, and then, the Mafia members, the inactive players, and the players who are failed targets of negotiation are removed from this set.

- **Detective:** There are some cases that the Detective must keep in mind. The players that are previously detected must not be detected again. The $po()$ and $ne()$ degrees specify these cases. Moreover, the winning strategy of the Detective is to detect the players who are probably in the Mafia team. $ex()$ is a good metric for finding the malicious players. Because the chance that the ones who vote for a removed player from the Townie team are in the Mafia team is high. $pr()$ is also another useful metric. The players who vote for the Proof, are probably Mafias. It must be mentioned that the Detective must not detect itself ($ro(4)$), or the inactive players ($in()$). In Algorithm 5 and Algorithm 6 we can see the winning strategies of the Detective in the case of using linear relation and reinforcement learning techniques, respectively. In Algorithm 5, a nested loop is used for finding the best coefficients of $ex()$ and $pr()$. They are found according to the

Algorithm 5 The winning strategy of the Detective using linear relation technique

Require: $\mathcal{B}_{DE}$, the belief parameters of Detective
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Ω, the ordered set of target candidates
  in, ex, pr, po, $ne \leftarrow \mathcal{B}_{DE}$
  $ro \leftarrow$ this player
  $mafias \leftarrow$ the Mafia team based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $best \leftarrow 0$
  $\Omega \leftarrow \{\}$
  for $i \in$ (-20 ... 20) **do**
    for $j \in$ (-20 ... 20) **do**
      $scores \leftarrow \{\}$
      for $k \in (1 \dots P)$ **do**
        $scores \leftarrow scores + \{k, i.ex[k] + j.pr[k]\}$
      $scores \leftarrow$ sorted $scores$
      $result \leftarrow 0$
      for $k \in (1 \dots |mafias|)$ **do**
        if $scores[k][1] \in mafias$ **then**
          $result \leftarrow result + 1$
      if $result > best$ or $\Omega = \{\}$ **then**
        $best \leftarrow result$
        $\Omega \leftarrow scores[1]$
  for $k \in (1 \dots P)$ **do**
    if $in[k] = 1$ or $po[k] = 1$ or $ne[k] = 1$ or $k \in ro$ **then**
      $\Omega \leftarrow \Omega$ - $\{k\}$
  return Ω

sorted set of the player's scores. The best result is achieved when the number of Mafia members at the beginning of this set has the highest value. Then, the players are sorted based on the found coefficients, and finally, the inactive players, the players who are previously detected, and the Detective itself is removed from the set. Algorithm 6 starts with a training phase that works based on the number of players at the beginning of the sorted list who is Mafia. When the training phase is over, the players are sorted based on the optimal scores, and then the players who must not be in the output settings are removed.

- **Doctor:** The Doctor must save the players who are probably shot during the night phase in order to help their team win. The first case that must be considered is that the Mafia team players must not be saved. Because they are not shot. $ex()$ and pr are good metrics for guessing the Mafias. On the other hand, the Mafia team often shoots a player who has correctly guessed the Mafia members. So, the Doctor can find the players who are on its side, and then save them ($si()$). Moreover, the Doctor must not save the inactive players, based on $in()$. According to these considerations, the winning strategies of the Doctor, using linear relation and reinforcement learning techniques, are presented in Algorithm 7 and Algorithm 8, respectively. A nested loop of three is considered for finding the coefficients of $ex()$, $pr()$, and $si()$. The best coefficients are the ones which lead to a sorted set that starts with the highest number of players who are shot. When the linear relation is found, the players are sorted based on the found scores, and finally, the inactive players are removed from the output set. In Algorithm 8, we have a training phase for finding the optimal scores. The number of players at the beginning of the sorted set, who are the target of the Godfather's shot, specifies the quality of the found scores. When the training phase is finished, the players are sorted, and then the inactive players are removed from the sorted set.

V. Network Modeling with Mafia

We can model a network and its security aspects with the proposed Mafia game model. The Townies are the legitimate hosts and the Mafias are the compromised nodes or the cyber adversaries (i.e. any malicious node). The interaction between the

Algorithm 6 The winning strategy of the Detective using reinforcement learning technique

Require: $\mathcal{B}_{DE}$, the belief parameters of Detective
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Ω, the ordered set of target candidates
  in, ex, pr, po, $ne \leftarrow \mathcal{B}_{DE}$
  $ro \leftarrow$ this player
  $mafias \leftarrow$ the Mafia team based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $model \leftarrow$ initiate the model
  for $i \in (0 \ldots 1000)$ **do**
    $scores \leftarrow$ generate the scores using $model$
    $result \leftarrow 0$
    for $k \in (1 \ldots |mafias|)$ **do**
      if $scores[k][1] \in mafias$ **then**
        $result \leftarrow result + 1$
    update $model$ based on $result$
  $scores \leftarrow$ generate the sorted scores using $model$
  $\Omega \leftarrow scores[1]$
  for $k \in (1 \ldots P)$ **do**
    if $in[k] = 1$ or $po[k] = 1$ or $ne[k] = 1$ or $k = ro$ **then**
      $\Omega \leftarrow \Omega$ - $\{k\}$
  return Ω

Algorithm 7 The winning strategy of the Doctor using linear relation technique

Require: $\mathcal{B}_{DO}$, the belief parameters of Doctor
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Γ, the ordered set of target candidates
  in, ex, pr, $si \leftarrow \mathcal{B}_{DO}$
  $shots \leftarrow$ the Mafia shots based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $best \leftarrow 0$
  $\Gamma \leftarrow \{\}$
  for $i \in (-20 \ldots 20)$ **do**
    for $j \in (-20 \ldots 20)$ **do**
      for $l \in (-20 \ldots 20)$ **do**
        $s \leftarrow \{\}$
        for $k \in (1 \ldots P)$ **do**
          $s \leftarrow s + \{k, i.ex[k] + j.pr[k] + l.si[k]\}$
        $s \leftarrow$ sorted s
        $result \leftarrow 0$
        for $k \in (1 \ldots P)$ **do**
          if $s[k][1] \in shots$ **then**
            $result \leftarrow result + 1$
          else
            break
        if $result > best$ or $\Psi = \{\}$ **then**
          $best \leftarrow result$
          $\Gamma \leftarrow s[1]$
  for $k \in (1 \ldots P)$ **do**
    if $in[k] = 1$ **then**
      $\Gamma \leftarrow \Gamma$ - $\{k\}$
  return Γ

players is the traffic which is forwarded through the network. The legitimate hosts, including defensive machines, do not know which nodes are malicious, and they attempt to find them. On the other hand, the malicious nodes try to create a united team in order to launch an attack against the other nodes. For example, when the malicious nodes establish multiple connections with a critical node to deny its availability, it is like when the mafias vote for a specific player to get rid of him/her in the game. Moreover, we can consider the Detective as an IDS, who detects the malicious nodes (i.e. the Mafias), or the Godfather as a botnet Command and Control server, who controls the botnet army (i.e. the whole Mafia team).

In this section, we describe the mapping of the Mafia game components onto the network components. There are some similarities between the Mafia game and a real network under a botnet attack, such as:

- The Mafia team cooperates with each other to cause the Townie team lose their power. The bots in a botnet also cooperate to reduce the power of the hosts in a network. For example, they try to perform a DDoS attack against one of the servers to make the network services unavailable.

Algorithm 8 The winning strategy of the Doctor using reinforcement learning technique

Require: $\mathcal{B}_{DO}$, the belief parameters of Doctor
Require: $\mathcal{D}$, the Mafia games dataset
Ensure: Γ, the ordered set of target candidates
  in, ex, pr, $si \leftarrow \mathcal{B}_{DO}$
  $shots \leftarrow$ the Mafia shots based on $\mathcal{D}$
  $P \leftarrow$ size of in
  $model \leftarrow$ initiate the model
  for $i \in (0 \ldots 1000)$ **do**
    $scores \leftarrow$ generate the scores using $model$
    $result \leftarrow 0$
    for $k \in (1 \ldots P)$ **do**
      if $scores[k][1] \in shots$ **then**
        $result \leftarrow result + 1$
      else
        break
    update $model$ based on $result$
  $scores \leftarrow$ generate the sorted scores using $model$
  $\Gamma \leftarrow scores[1]$
  for $k \in (1 \ldots P)$ **do**
    if $in[k] = 1$ **then**
      $\Gamma \leftarrow \Gamma$ - $\{k\}$
  return Γ

- The Mafia team has a leader (i.e. Godfather), who is hard to detect and also manages the team to perform coordinated tasks. A botnet also contains a command and control server that manages the bots to launch the final attack, and it is also hard to detect.
- The players in the Mafia team know each other, while the Townies are not sure about the identity of the other Townies. In a real network, it is the same. The components in a botnet know each other, while in general, the legitimate hosts are not sure which of the other hosts are trusted.
- Negotiating with a Townie through a night phase is similar to loading the botnet script on a normal host to convert it into a bot. In both processes, a good party becomes a bad one. Moreover, negotiation in a Mafia game is done only with the Townies. In a network is similar, and the security tools, such as IDS, are safe enough to not become a bot.
- Voting to a player in a day phase in order to remove it from the game is similar to sending flooding traffic toward a host in order to cause it become unavailable.
- Shooting a player in a night phase in order to remove it from the game is similar to compromising a host with a malware by the adversary in order to make it lose its functionalities in the network.
- A Narrator controls the game flow and manages the shifts between the phases. In an SDN there is also a central component called the controller, which monitors the network, and manages the traffic. In the networks other than SDNs, the logic and protocols of the network are the Narrator.

There are also some other similarities between the game roles and network components as follows (Figure 1):

- **A command and control server plays the role of Godfather:** The command and control server, or generally the adversary, is a member of the botnet, who does not take a role in the final attack of the botnet. However, it is responsible for commanding the bots or the loader to perform some tasks. Since this member is not involved in the final malicious activities, it is hard to detect. These are the features of a Godfather in the Mafia team.
- **A botnet script loader plays the role of Cultafia:** The script loader is responsible for loading a malicious script on the normal hosts, that can recruit them as a bot. However, it is under the control of the command and control server. This behavior matches the behavior of a Cultafia in the Mafia team.
- **A bot plays the role of Mafia:** The bots in a botnet are under the command of the adversary, in order to launch the final attack. The target is specified by the command and control server, and the bots obey it. The activities of the Mafia players are similar to the bots.
- **An IDS plays the role of Detective:** The IDS in a network, wastes its resources on accurately detecting a specific node. For example, an IDS knows malicious patterns, and then monitors the traffic from a suspicious node to check its activity patterns. This responsibility is similar to the tasks of a Detective in a Mafia game. Moreover, the diagnosis of an IDS is not always correct. In other words, the false positive and false negative rates of the IDSs are not always zero. This characteristic is similar to the Detective. When the Detective receives a thumbs down from the Narrator, the target player is not definitely on the Townie team. There is also another similarity between the IDS and the Detective. The data collected by the IDS can be used for improving the other security tools. The facts collected by the Detective are also used by the other Townie team players to effectively detect the Mafia members.
- **An anti-malware plays the role of Doctor:** The anti-malware can protect the hosts from being infected. In other words,

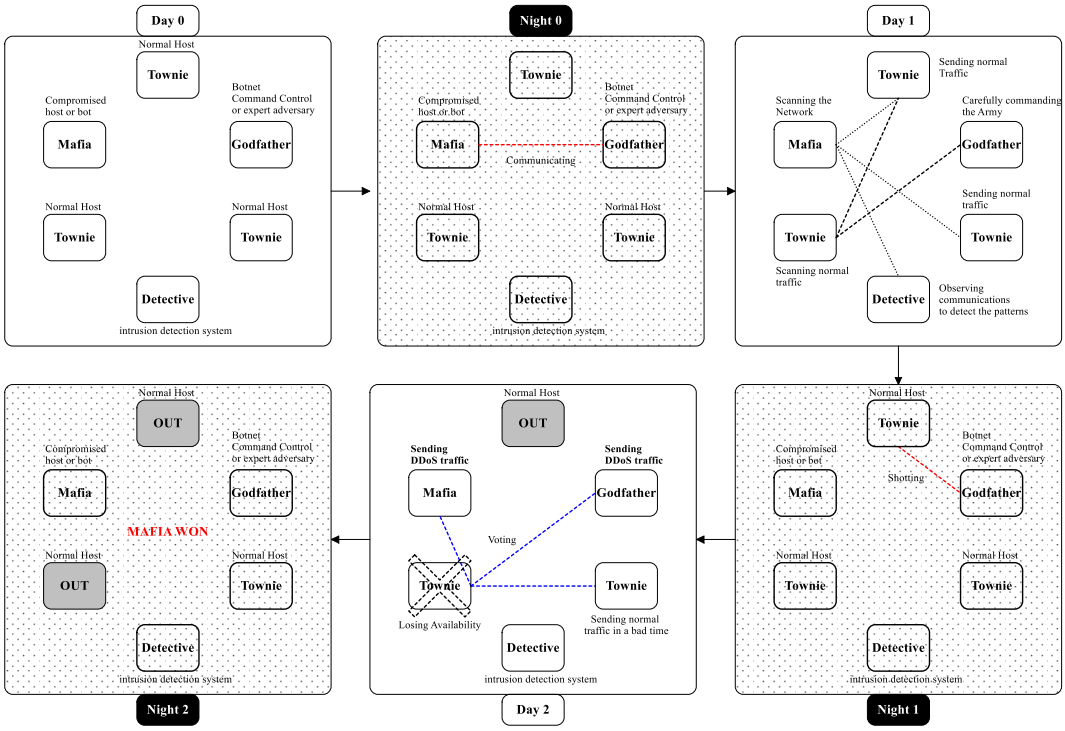


Fig. 3. A sample Mafia game with a Godfather, a Mafia, a Detective, and three Townies, and its equivalent in a computer network (More examples in Appendix 3)

the anti-malware saves the hosts from losing their functionalities. The doctor in a Mafia game performs a similar task.

- **A honeypot plays the role of Proof:** A honeypot in a network does not have a productive value. So the legitimate host rarely connects with it. The connections with the honeypots are probably established by the illegal nodes (e.g. the bots). Proof is somehow the same. When the players vote for Proof, they become suspicious. The malicious nodes are afraid of communicating with the honeypots because this connection leads to their detection. The Mafias are also concerned about voting for Proof. Moreover, a honeypot has powerful security mechanisms, and it is protected against the cyber attacks, like DDoS. So, if flooded traffic is sent toward a honeypot, it does not lose its functionalities. A Proof in the Mafia game is the same. When numerous votes are against the Proof, this player will not be removed from the game.
- **A normal host plays the role of Townie:** The normal hosts do not have a security value. However, their connections help the defenders in finding the legal traffic patterns. Townies are doing the same in a Mafia game.

It must be noted that all the activities that can be observed by simply monitoring the traffic are considered as the activities during the day phase, and the other covert activities are equivalent to the night activities. For example, sending a traffic toward a host is like the player's conversation during the day phase, while compromising a host is equivalent to a shot or negotiation in the night phase. The voting at the end of each day may be considered as some heavy traffic going to a host (e.g. in a DDoS attack), and if that player is removed from the game due to a high number of votes, it is considered equivalent to the success of the DDoS attack.

According to these similarities, we can apply the strategies of detecting the Mafia players (i.e. the Detective's strategies) to a real network in order to detect the malicious nodes. These strategies help the network defender to effectively detect the bots or any malicious nodes, and then block them. The main goal of modeling a network with a Mafia game is to predict the malicious nodes by their connections. Therefore, we can say that the connections between the players are more important than the internal activities and detailed features of each node/player. For example, if a specific node establishes several connections to a server, and these connections lead to a DDoS attack, that node may probably be a malicious one. However, we cannot certainly consider that node as malicious. Again, suppose that this node is involved in a DDoS attack against another server. So, it is more probable to be malicious than before. In summary, the major goal we are concentrating on is the connections. In Mafia games, the connections are the most important facts that lead to detecting the Mafia team.

Figure 3 shows a sample network which is modeled by a Mafia game. The actions of each component are shown in each phase. This network contains a command and control server, a bot, an IDS, and three normal hosts, which is equivalent to a Mafia game with a Godfather, a Mafia, a Detective, and three Townies. Night 0 is the time when the command and control server communicates with the bots in order to introduce them to the other botnet members. In a Mafia game, night 0 is the time when players have received their roles from the Narrator, and the Mafia team open their eyes to identify each other. In day 1,

the legitimate hosts send normal traffic to their desired destination, while the botnet members do their job. The bot scans the network for sending the useful information to the command and control server, and the command and control server signals it with its command. On the other hand, the IDS observes the traffic to monitor the patterns. In the equivalent Mafia game, the Townies vote for some specific players, while the Mafia team has a strategy in its votes. The Detective also pays attention to the other players' votes to find more information about them. In night 1, the adversary who is located on the command and control server, compromises one of the normal hosts, and removes it from the network. This is equivalent to shooting one of the Townies in the Mafia game. In day 1, the command and control server signals the bot to send flooded traffic toward a specific host. On the other hand, another host is also requesting for a service from that host. As a result, a DDoS attack may occur, and the normal host will be removed from the network. This is equivalent to voting to a specific Townie and removing it from the game. At this time, two out of six nodes in the network are removed, and two of them are also malicious nodes. In other words, half of the networks are malicious. So the entire network does not have enough functionality. This is the condition in which the Mafia team wins.

VI. Evaluation

To evaluate the proposed model and the suggested strategy, we should consider two aspects: (1) checking the suggested strategy with the Mafia game datasets and (2) checking the suggested strategy with a real network. The first aspect makes sure that the suggested strategy for detecting the Mafia players is effective, and the second one is for evaluating the performance of modeling a network with the Mafia game. One way to evaluate this effectiveness is to apply the suggested strategies on a real game/network, and then investigate the final result, which is the winning of the Mafia or Townie team. However, we do not access an unbiased group of players, by which we can apply the strategies and check the final winner, and additionally it is out of the scope of this paper to design an agent that automatically plays the game. Moreover, different factors may affect the final result in a real Mafia game, such as human personality behaviors and players' body language. As a result, we have extended the meaning of two common metrics in security fields, True Positive Rate (TPR) and True Negative Rate (TNR), for each of the strategies, and then evaluate their efficiency using these metrics.

For the first aspect, we need the dataset of real-world Mafia games [More Details MAFIA dataset in Appendix 1, http://mosaic-lab.org/data-set/mafia-dataset.zip]. As a result, we have watched 7 hours of the Mafia game played by Iranian players, which are available on Youtube [56], and then collected their major data to create a Mafia dataset. There are initially ten players in these games, three of which are in the Mafia team. We have compared the evaluation results with the related strategies that are suggested by the previous researches. For the second aspect, we have simulated different networks which are under the attack of the Mirai [57] botnet.

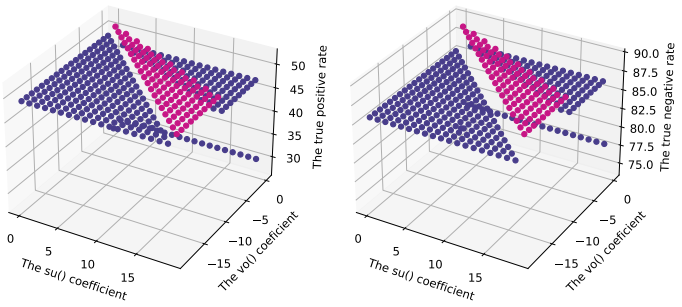
The two suggested techniques are evaluated in this section. For the reinforcement learning technique, we have used a neural network with two hidden layers containing 256 neurons. The problem model that is passed to the agent is a game of sorting different cards, each of which has specific parameters. For example, in a game with P players, the agent that tries to find the best strategy for the Godfather gets P cards, and each card has three values equivalent to the $ma()$, $su()$, and vo degrees. These cards represent each of the players. When the agent sorts the cards, the first card will be suggested to the Godfather as its target. The action space of the agent contains only two actions. The first action, 0, means the i^{th} card must not be replaced by the j^{th} card. The second action, 1, means that the location of the i^{th} and j^{th} cards must be changed. During the game steps, the values of i and j are changed so that at the end of the game, all the cards are considered in the sorting process. 85% of the games in the dataset are used for the training phase, and the remained 15% are used for testing the learning performance.

In the remaining of this section, the details of the proposed model evaluation, including Godfather's, Cultafia's, Detective's, Doctor's, and network strategies efficiency, are mentioned. The applicability of the proposed Mafia game model to real-world network security relies on accurately mapping adversarial and defensive roles. To further support this analogy, additional evidence is necessary to validate claims regarding the difficulty of identifying botnet command-and-control (C2) servers. Previous research has demonstrated that botnet operators employ fast-flux networks, domain generation algorithms (DGA), and peer-to-peer (P2P) architectures to evade detection, making traditional tracking methods less effective. These techniques parallel the Mafia game strategy, where the Godfather avoids exposure by blending in and manipulating other players. Providing empirical case studies or referencing established botnet detection challenges will reinforce this claim, demonstrating the real-world significance of the proposed model.

A. Evaluating Godfather's strategy

Shooting the players during the night phases is the action of the Godfather in a Mafia game. Using the suggested strategies, Godfather can effectively shoot the players. We define the values of TPR and TNR for the Godfather as Equation 39 and Equation 40, respectively.

$$\text{TPR}_{GO} = \frac{\text{The no. of targets who may be removed}}{\text{The total no. of targets}} \tag{39}$$



(a) TPR (b) TNR

Fig. 4. The value of TPR and TNR for the Godfather's strategy using the linear relation technique based on the Mafia game dataset according to the different values of $su()$ and $vo()$ coefficients

$$\text{TNR}_{GO} = \frac{\text{The no. of non-targets who may not be removed}}{\text{The total no. of non-targets}} \tag{40}$$

For example, suppose that in a specific night, there are six Townie members, and up to two players can be shot. Now, the Godfathers sorts the players, and the first two players are considered as its targets, and the other four players are non-targets. If both of the first two players are possible to be removed, the TPR value is 100%. Moreover, if three of the four non-target players are not possible to be removed (e.g. because of the Doctor saving or being the Proof), the TNR value is 75%.

According to the defined TPR and TNR for the Godfather, $tpr\_go(i)$ and $tnr\_go(i)$, which are the TPR and TNR values in the i^{th} night, can be calculated as Equation 41 and Equation 42, respectively.

$$tpr\_go(i) = \frac{\sum_{j=1}^{\alpha'} C_{\text{shot}}(i,j)}{\alpha'} \tag{41}$$

$$tnr\_go(i) = \frac{\sum_{j=\alpha'+1}^{|\Theta|} (1 - C_{\text{shot}}(i,j))}{|\Theta| - \alpha'} \tag{42}$$

Now we can evaluate the Godfather's strategies, both with the linear relation and reinforcement learning techniques, using the defined TPR and TNR metrics.

Figure 4 illustrates the value of TPR and TNR for the Godfather's strategy using the linear relation technique. We can see the impact of different degrees' coefficients on the final result. The point with the light color in these two graphs are the ones with the highest value of TPR/TNR comparing the other points. The highest values of TPR and TNR are about 51% and 89%, and these values are obtained when the coefficients of $su()$ and $vo()$ are a non-zero positive and negative number, respectively, and also the coefficient of $su()$ is greater than or equal to the absolute value of the $vo()$ coefficient, but not higher than or equal to the double of $vo()$ coefficient. This result indicates that the $su()$ degree is more important than $vo()$, and in the sorting process, the players with higher value of $su()$ must be placed at the end of the sorted set. This is reasonable because the players who survived the previous night are more likely to survive the next night. The other point is that the importance of $su()$ is not two times higher than $vo()$.

For comparing the results of the linear relation and reinforcement learning techniques, we have reported each technique's best results. One of the best coefficients for $su()$ and $vo()$ are 1 and -1 [58, 59]. So, we have used the linear relation of $su() - vo()$ for this part. Figure 5 shows this comparison. The maximum TPR and TNR values of the reinforcement learning technique are 100%, and both of them are higher than the linear relation results. So, we can say that the reinforcement technique can predict the best strategy, based on both TPR and TNR. We have also compared these results with the related previous strategies. We can see that none of them outperform our proposed reinforcement learning technique. The results show that the TPR value of "Previous Strategy 2" is higher than the TPR value of our proposed linear relation technique. In other cases, the proposed linear relation technique is more efficient than the other previous strategies. The average results show that the Godfather's shot using our suggested strategy are 15.5% better than the previous strategies.

Another point about the results is the higher value of TNR compared with TPR. This is because of the small size of the targets in comparison with the non-targets. In other words, finding the targets who can successfully be shot is harder than finding the players who may not be removed.

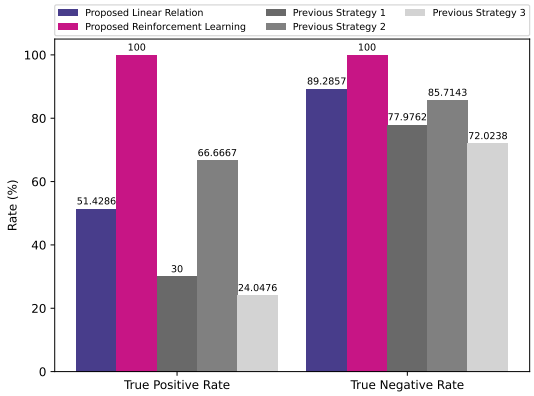


Fig. 5. The values of TPR and TNR by applying the suggested Godfather's strategies on the Mafia dataset using the linear relation and reinforcement learning techniques

B. Evaluating Cultafia's strategy

The Cultafia negotiates with one of the Townies to make the Mafia team bigger. The Cultafia can use the suggested strategies to negotiate with a higher success chance. To evaluate the suggested strategy, we define the TPR and TNR values for the Culatafia as mentioned in Equation 43 and Equation 44.

$$\text{TPR}_{CU} = \frac{\text{The no. of targets who may be negotiated}}{\text{The total no. of targets}} \tag{43}$$

$$\text{TNR}_{CU} = \frac{\text{The no. of non-targets who may not be negotiated}}{\text{The total no. of non-targets}} \tag{44}$$

We give an example for these metrics. Suppose that six Townie members are active in the game, and there are only two players among them who can be negotiated. If only one of the first two players in the Cultafia's sorted set is a Townie, the value of TPR is 50%. If only one of the remained players in the set can be negotiated, the TNR value will also be 75%.

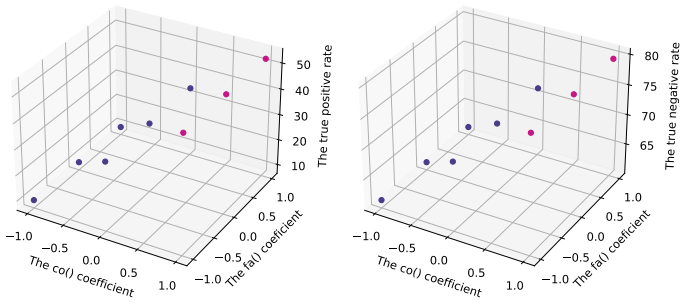
If we name the value of Cultafia's TPR and TNR at the i^{th} night as $tpr\_cu(i)$ and $tnr\_cu(i)$, these values can be calculated based on Equation 45 and Equation 46, respectively.

$$tpr\_cu(i) = \frac{\sum_{j=1}^{|ro^i(7)|} C_{\text{changed}}(i,j)}{|ro^i(7)|} \tag{45}$$

$$tnr\_cu(i) = \frac{\sum_{j=|ro^i(7)|+1}^{|\Psi|} (1 - C_{\text{changed}}(i,j))}{|\Psi| - |ro^i(7)|} \tag{46}$$

Three metrics are used in the belief function of the Cultafia, which are $ma()$, $co()$, and $fa()$. However, the $ma()$ and $fa()$ degrees are for removing some players from the targets set, and there is no need for considering them in a linear relation. As a result, the only important aspect about the coefficient of $co()$ is its sign. As we can see in the results of Figure 6, the coefficient of $co()$ must be positive, and this is reasonable. The dots with the light color are related to the maximum value of TPR/TNR. At the first spot, it seems that the positive value of the $co()$ coefficient is not reasonable. Because the players with a night-phase ability are more accurate in detecting the Mafias. However, the collected dataset reveals something different. The players with no abilities are more accurate. Because they have no extra information rather than the votes, and they behave more wisely than the players with night roles. The other coefficients except 1, 0, and -1 are not investigated because when a single parameter is in a linear relation, only the sign of the coefficient is important. The other coefficients result in a similar output. The highest reported values of TPR and TNR are 52% and 79%, respectively.

To compare the performance of the linear relation and reinforcement learning techniques, we have used $+co()$ as the linear equation to achieve the highest performance. The results are shown in Figure 7. The greatest TPR and TNR values of the reinforcement learning technique are 100%, and we conclude that in the presence of a dataset, the reinforcement learning technique outperforms the linear relation technique. Moreover, the $co()$ degree is a sufficient metric for detecting the players with no specific roles. However, we predict that the agent may be confused when it is trained with a larger dataset. So, the high value of TPR and TNR in the reinforcement learning technique does not show that the suggested strategy does not have



(a) TPR (b) TNR

Fig. 6. The value of TPR and TNR for the Cultafia's strategy using the linear relation technique based on the Mafia game dataset according to the different values of $co()$ and $fa()$ coefficients

Fig. 7. The values of TPR and TNR by applying the suggested Cultafia's strategies on the Mafia dataset using the linear relation and reinforcement learning techniques

any false detections. This result is just for the current dataset. Since none of the previous researches have considered the role of Cultafia, we have no comparisons here.

Similar to the results of the Godfather's strategy evaluation, the TPR values are lower then the TNR values for Cultafia's strategy. This is due to a similar fact, which is related to the target set size.

C. Evaluating Detective's strategy

The responsibility of the Detective is to detect the Mafia members. In our suggested strategy, the Detective can find them with a higher chance. The defined TPR and TNR values for evaluating the suggested strategy are presented in Equation 47 and Equation 48.

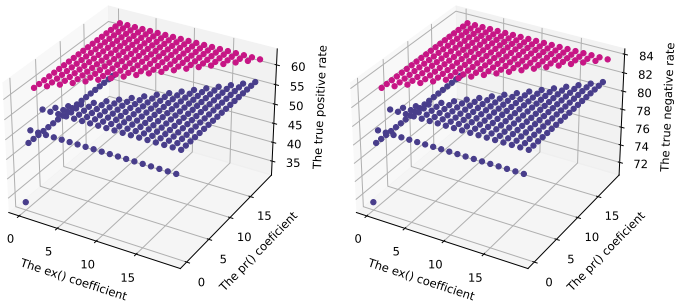
$$\text{TPR}_{CU} = \frac{\text{The no. of targets who are Mafia members}}{\text{The total no. of targets}} \tag{47}$$

$$\text{TNR}_{CU} = \frac{\text{The no. of non-targets who are Townie members}}{\text{The total no. of non-targets}} \tag{48}$$

Based on these metrics, we give an example. Consider a game with 3 Mafias and 7 Townies. The sorted set generated by the Detective's belief function contains 9 players, because the Detective itself is not in this set. According to the Detective's opinion, the first three members of this set are Mafias and the others are Townies. Hence, if only two of the first three players are Mafias, the values of TPR and TNR are 66% and 83%, respectively.

Assume that $tpr\_de(i)$ and $tnr\_de(i)$ are the Detective's TPR and TNR at the i^{th} night. These values are calculated regarding Equation 49 and Equation 50, respectively.

$$tpr\_de(i) = \frac{\sum_{j=1}^{|ro_m(i)|-1} C_{\text{up}}(i,j)}{|ro_m(i)| - 1} \tag{49}$$



(a) TPR (b) TNR

Fig. 8. The value of TPR and TNR for the Detective's strategy using the linear relation technique based on the Mafia game dataset according to the different values of $ex()$ and $pr()$ coefficients

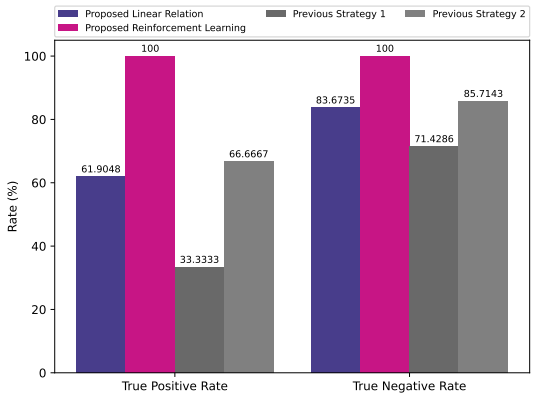
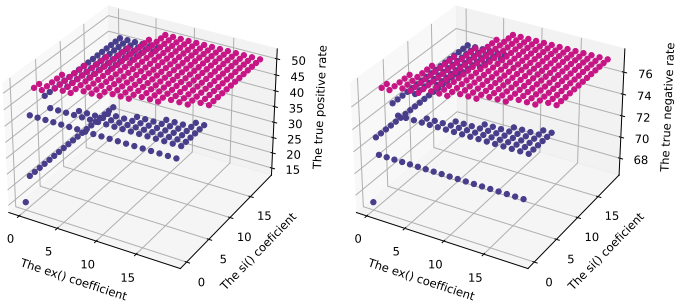


Fig. 9. The values of TPR and TNR by applying the suggested Detective's strategies on the Mafia dataset using the linear relation and reinforcement learning techniques

$$tnr\_de(i) = \frac{\sum_{j=|ro_m(i)|}^{|\Omega|} (1 - C_{\text{up}}(i,j))}{|\Omega| - |ro_m(i)| + 1} \tag{50}$$

In Figure 8, the different coefficients of $ex()$ and $pr()$ and their impact on the TPR and TNR values are shown. The best results (i.e. the points with the light color) are obtained where both coefficients are non-zero positive numbers and the coefficient of $pr()$ is greater than or equal to the coefficient of $ex()$. This means that the importance of the $pr()$ degree is higher than the $ex()$ degree for detecting the Mafia players. In other words, the players who have voted for a revealed Proof are more likely to be a Mafia rather than the players who vote for other removed Townies. The greatest achieved values for TPR and TNR are about 61% and 83%, respectively. Another important point about this graph is where the coefficients are zero. In this case, the values of TPR and TNR are low, which that indicates both of the $ex()$ and the $pr()$ degrees must be considered in the winning strategy of the Detective. Since the players who have voted one of the Townie members are more likely to be a Mafia, the positive sign of the coefficients is reasonable.

Again, for comparing the linear relation and reinforcement learning techniques, the best obtained results are reported. Hence, for the comparison of this part, we have used the equation of $ex() + 2pr()$ as the linear relation. The comparison is illustrated in Figure 9. Based on these results, the reinforcement learning technique can detect 100% of the Mafia and Townie players, using the suggested strategy. On the other hand, the linear relation technique can detect about 61% and 83% of the Mafias and Townies, respectively. We can conclude that the reinforcement technique works better than the linear relation technique when a dataset is available. The other point about these results is that, due to the greater values of TNR against TPR, the suggested strategy works better in detecting the Townies compared with detecting the Mafias. Similar to the high results achieved by the reinforcement learning technique in Cultafia's strategy, for the Detective's strategy we cannot say that the strategy can absolutely detect the Mafias in every game. The results are currently reliable for our used dataset. We have also compared the results with the previous related strategies. The proposed reinforcement learning technique has the best results among the other strategies, and the second place belongs to "Previous Strategy 2". On average, the detection rate of our suggested strategy for the Detective is 11% higher than the previous strategies.



(a) TPR (b) TNR

Fig. 10. The value of TPR and TNR for the Doctor's strategy using the linear relation technique based on the Mafia game dataset according to the different values of $ex()$ and $si()$ coefficients

D. Evaluating Doctor's strategy

The suggested strategy of saving the player helps the Doctor in selecting its targets. To evaluate it, we have defined the values of TPR and TNR for the Doctor (Equation 51 and Equation 52).

$$\text{TPR}_{CU} = \frac{\text{The no. of targets who are shot}}{\text{The total no. of targets}} \tag{51}$$

$$\text{TNR}_{CU} = \frac{\text{The no. of non-targets who are not shot}}{\text{The total no. of non-targets}} \tag{52}$$

Suppose that there are six players, except the Doctor, in the game, and up to two players can be shot. The Doctor sorts the players, and then selects the first two players of the sorted sets as its targets. The remaining four players are considered as its non-targets. If only one of the first two players are shot, and the other shot target is selected from the next four players, the values of TPR and TNR are 50% and 75%, respectively.

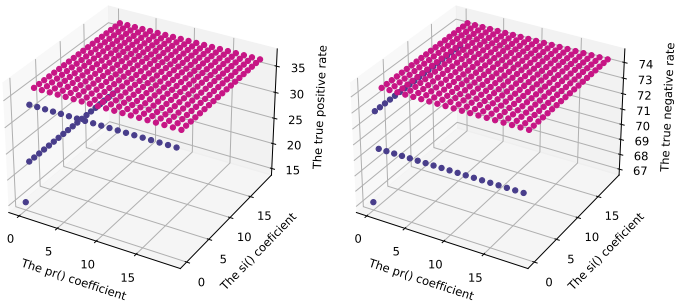
If we call the value of Doctor's TPR and TNR at the i^{th} night as $tpr\_do(i)$ and $tnr\_do(i)$, we can calculate them using Equation 53 and Equation 54, respectively.

$$tpr\_do(i) = \frac{\sum_{j=1}^{\alpha'} (1 - C_{\text{shot}}(i,j))}{\alpha'} \tag{53}$$

$$tnr\_do(i) = \frac{\sum_{j=\alpha'+1}^{|\Gamma|} C_{\text{shot}}(i,j)}{|\Gamma| - \alpha'} \tag{54}$$

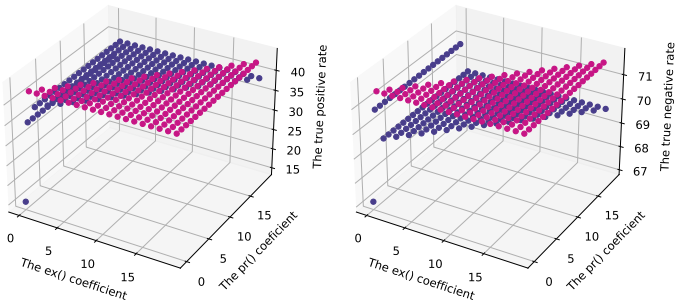
Three degrees, $ex()$, $pr()$, and $si()$, are considered in the definition of the Doctor's belief function. Hence, the different values of three coefficients must be considered in evaluating the linear relation. The obtained results show that fixing the coefficients of $ex()$, $pr()$, and $si()$ to 2, 1, and 3, leads to the best results. However, since showing all the three coefficients in a 3D space is not possible, we have shown them in separate graphs, where the missed degree has the coefficient of zero. Figure 10 shows the impact of $ex()$ and $si()$ coefficients on the TPR and TNR values, where the $pr()$ coefficient is fixed. The results show that the coefficient of $ex()$ and $si()$ must be both non-zero positive numbers, where the $si()$ coefficient is greater than or equal to the $si()$ coefficient, but not higher than its triple. This means the importance of $si()$ is higher than $ex()$ when $pr()$ is not considered, but not without limitations. Figure 11 illustrates the impact of $pr()$ and $si()$ coefficient on the final result. We can see that when $ex()$ is not considered, the TPR and TNR are not in their highest value, and the difference between the importance of $pr()$ and $si()$ cannot be investigated. Finally, Figure 12 presents the impact of $ex()$ and $pr()$ coefficients. Both of them must be non-zero positive numbers, where the coefficient of $ex()$ must be greater than or equal to $pr()$ coefficient. Again the highest value is not achieved, but we can see that the results are the reverse form of the Detective's result. In other words, the Detective must find the suspicious players, and they are in Mafia team. But the Doctor must find the players who are suspicious, but they are actually a Townie. Because the Mafia team shot the players who are suspicious, and the probability of their saving is low.

The overall results show that the highest achieved TPR and TNR are 50% and 77%, where the linear relation is $2ex() + pr() + 3si()$. This equation is used for comparing the linear relation and reinforcement learning techniques, as it is shown in Figure 13. Using reinforcement learning technique, the Doctor can correctly select all (i.e. TPR and TNR values of 100%) the players for saving them. It must be noted again that this result is based on the current dataset. The results also show that



(a) TPR (b) TNR

Fig. 11. The value of TPR and TNR for the Doctor's strategy using the linear relation technique based on the Mafia game dataset according to the different values of $pr()$ and $si()$ coefficients



(a) TPR (b) TNR

Fig. 12. The value of TPR and TNR for the Doctor's strategy using the linear relation technique based on the Mafia game dataset according to the different values of $ex()$ and $pr()$ coefficients

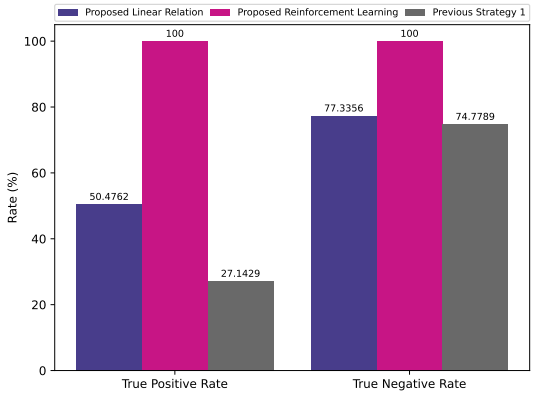


Fig. 13. The values of TPR and TNR by applying the suggested Doctor's strategies on the Mafia dataset using the linear relation and reinforcement learning techniques

both linear relation and reinforcement learning techniques outperform "Previous Strategy 1". The average results show that the saving rate of the Doctor in our suggested strategy is 10% higher than the previous strategy.

E. Evaluating real networks strategy

After evaluating the performance of the suggested strategy in Mafia games, it is required to also evaluate it in real networks. We have simulated different networks in Mininet, and then we have propagated the Mirai bots in these networks. Mirai has three main components: the command and control server, the loader, and the bots. The adversary is located on the command and control server, and commands the bots to launch a DDoS attack against a specific target. The initial bots scan the network, and whenever they found the username and password pair of a host, they send it to the loader. The loader then loads the malicious script on the victim hosts. In the simulated networks, single nodes are dedicated for the IDS, honeypot, command and control, and loader. The number of initial bots and legal hosts varies in different networks. If b is the number of initial

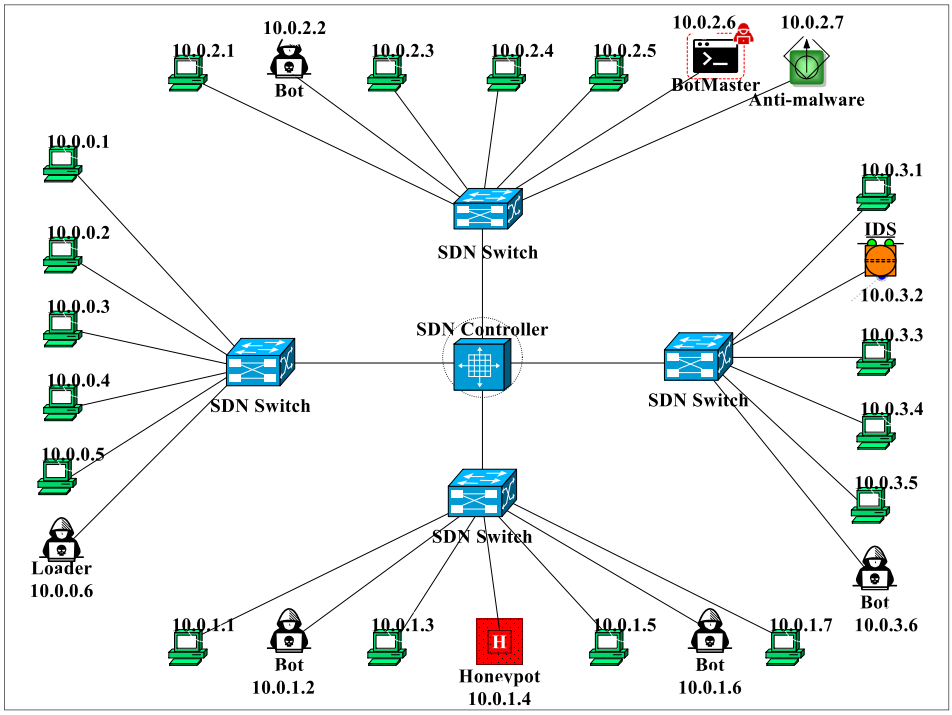


Fig. 14. The simulated network topology with four initial Mirai bots

bots, the number of legal hosts is $4b + 1$, where b varies from 1 to 10. The network topology with four initial bots is shown in Figure 14. The IDS can detect up to b nodes, and the username and passwords of each host are selected among a set of four pairs.

To calculate the detection efficiency, we have reported the TPR and TNR results based on the common definition of these two metrics. TPR is a metric for evaluating the power of detecting the malicious nodes or bots in the network. This metric can be calculated as Equation 55.

$$\text{TPR}_{net} = \frac{\text{The no. of detected bots}}{\text{The total no. of detected nodes}} \tag{55}$$

On the other hand, TNR shows the ability of detecting the legal nodes in the network. TNR is calculated based on Equation 56.

$$\text{TNR}_{net} = \frac{\text{The no. not detected legal hosts}}{\text{The total no. of not detected nodes}} \tag{56}$$

Figure 15 shows the TPR and TNR values of applying the suggested strategy of the Detective on the data collected from the simulated networks in Mininet. The results of the linear relation technique show that the detection rate, both for the bots and legal hosts, increases as the number of initial bots gets higher. We can see that when the number of bots are 10, the linear relation technique can detect up to 83% and 95% of the bots and legal hosts, respectively. Moreover, we can see that as the number of initial bots increases, the detection rate of the linear relation technique outperforms the reinforcement learning technique. This shows the power of the linear technique in large-scale networks. Large-scale networks have more connections, and since our strategy is based on the connections, it works better in these networks. On the other hand, we can see that the detection rate of the reinforcement learning technique is 100% when the number of initial bots is one or two. This is because the reinforcement learning agent is trained with the Mafia game dataset, which includes only a single Mafia. Therefore, this is just that it works better in the cases similar to the Mafia dataset. In general, the average results of both techniques show that the suggested strategy has the TPR and TNR values of about 71% and 91%, which are an acceptable detection rates. There is also another point about the results. The gap between the values of TPR and TNR in Figure 15 is similar to the graphs related to Mafia roles strategies evaluations. This similarity is a satisfactory result for claiming that the Mafia game is a perfect candidate for modeling the security problems of the real-world networks.

Model Performance Comparison and Analysis

In this section, we comprehensively evaluated multiple machine learning and deep learning models to assess their effectiveness in classifying network traffic as benign or malicious. The models were tested on both simulated data and the real-world CIC-

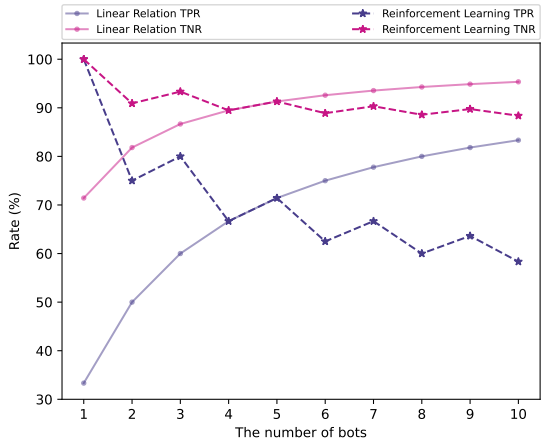


Fig. 15. The values of TPR and TNR by applying the suggested Detective's strategy on the simulated networks in Mininet using the linear relation and reinforcement learning techniques

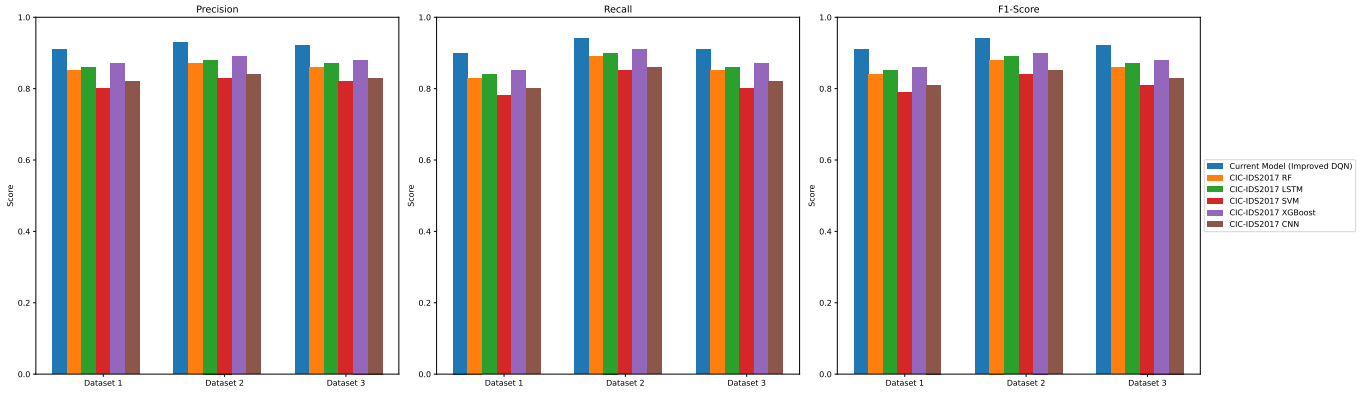


Fig. 16. Comprehensive Comparison of Model Performance Metrics Across Methods and Real-World CIC-IDS2017 Dataset.

IDS2017 dataset [1], a widely recognized benchmark in network intrusion detection.

The comparative analysis reveals that ensemble methods like Random Forest and XGBoost significantly outperform traditional machine learning and deep learning models in network intrusion detection. Although DQN models benefit from optimization, their performance is still poor compared to real-world applications of tree-based algorithms and advanced boost techniques. Future research could explore hybrid models that combine the strengths of various approaches and further enhancements in feature engineering and real-time detection capabilities (More Details CIC-IDS2017 dataset in Appendix 2).

Figure 16 presents a comparative analysis of different machine learning models in terms of precision, recall, and F1 score in three datasets. The Current Model (Improved DQN) consistently outperforms all other models, achieving the highest scores in all three metrics. Specifically, it achieves precision values of 0.91, 0.93, and 0.92, recall values of 0.90, 0.94, and 0.91, and F1 scores of 0.91, 0.94, and 0.92. These results indicate that the improved DQN model demonstrates superior predictive capability and robustness compared to other state-of-the-art models, including Random Forest (RF), Long Short-Term Memory (LSTM), Support Vector Machine (SVM), XGBoost, and Convolutional Neural Network (CNN). The results also highlight that while traditional models such as SVM and CNN perform adequately, their recall values tend to be lower, which could lead to higher false negative rates. The bar chart clearly illustrates the consistent performance gains achieved by the improved DQN approach, making it a promising candidate for real-world deployment in cybersecurity applications.

In Figure 17, This presents a visualization of the training performance of the Deep Q-Network (DQN) agent. The green line represents the total reward accumulated over 50 episodes, demonstrating an increasing trend as the model learns better policies. The blue dashed line represents the epsilon decay, which decreases over time, reducing the agent's exploration in favor of exploitation. This trade-off ensures a balance between discovering new strategies and refining existing ones.

In Figure 18, we present a comparative performance analysis of various machine learning models, including Random Forest, SVM, XGBoost, LSTM, CNN, and our proposed model, the Improved DQN, applied to the simulated CIC-IDS2017 dataset. The results show that the Improved DQN consistently outperforms all other models in terms of precision, recall, and F1 score

[1] https://www.kaggle.com/datasets/dhoogla/cicids2017?resource=download

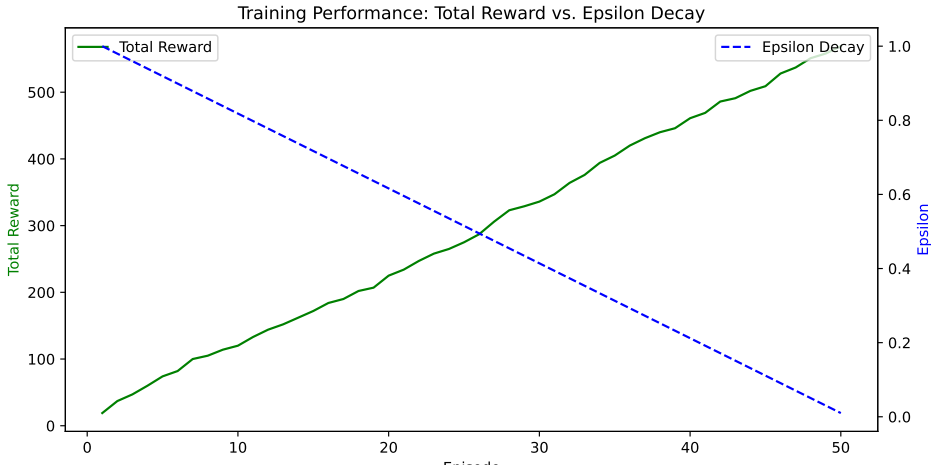


Fig. 17. A training performance of the Deep Q-Network (DQN) agent.

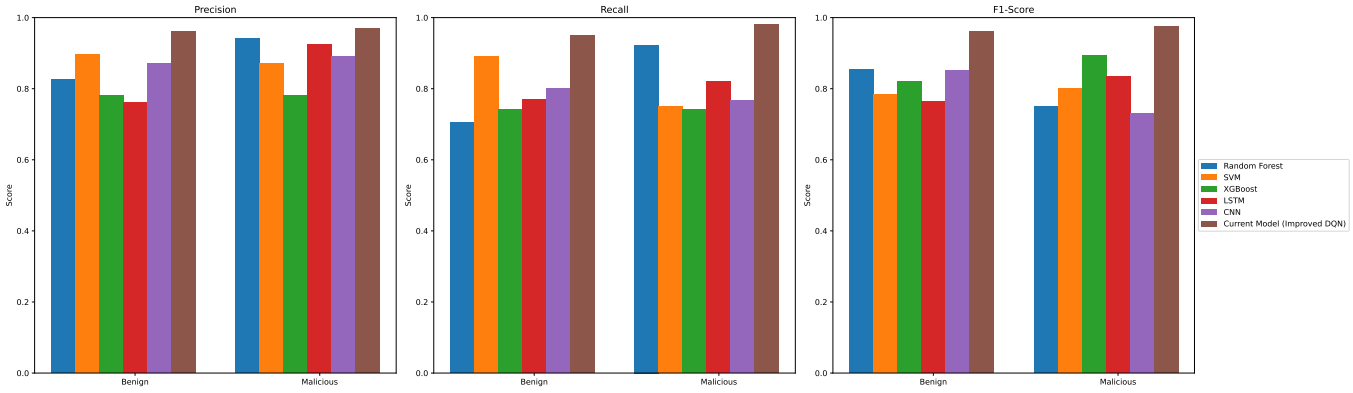


Fig. 18. Model Performance on CIC-IDS2017 Dataset with Improved DQN

for both benign and malicious classifications. Notably, the Improved DQN achieves the highest recall (0.98) and F1 score (0.975) for detecting malicious traffic, demonstrating its superior ability to identify network threats while minimizing false negatives. While XGBoost and CNN also yield competitive results, they have slightly lower recall values, indicating a potential trade-off between detection sensitivity and classification stability. Traditional models like SVM and Random Forest perform well in precision but fall short in overall balanced performance. This comparison underscores the effectiveness of the Improved DQN model as a robust and accurate solution for detecting network security threats.

Suggested Detective's Strategy Current Model (DQN)

The initial Deep Q-Network (DQN) model demonstrated baseline performance with precision, recall, and F1-scores around 50% for both benign and malicious classes. This performance suggests that the model struggled to effectively distinguish between normal and malicious network behavior, performing slightly better than random guessing. The current model's limitations could be attributed to insufficient hyperparameter tuning, a simplistic reward structure in the custom environment, and potential inadequacies in feature selection. The Figure 19 presents a comparative analysis of multiple machine learning models—Random Forest, SVM, XGBoost, LSTM, CNN—and a Suggested Detective's Strategy. The Suggested Strategy, highlighted prominently in gold, demonstrates superior results across all metrics, indicating its enhanced capability in accurately detecting cyber threats while minimizing false positives and negatives.

Improved DQN Model

Significant improvements were observed after optimizing the DQN model through hyperparameter tuning, architectural enhancements, and reward restructuring. The optimized model achieved a precision of 65% for benign traffic and 70% for malicious traffic, with corresponding recall values of 60% and 75%. These results highlight the impact of refining learning rates, adjusting exploration strategies, and incorporating more robust feature engineering techniques, leading to a notable increase in the model's ability to classify network data correctly.

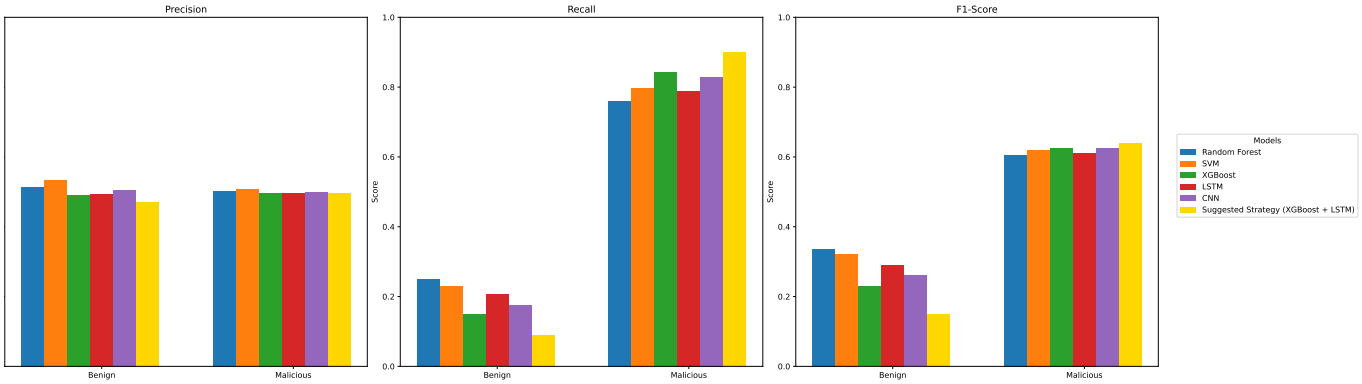


Fig. 19. Suggested Detective's Strategy Performance on Simulated CIC-IDS2017 Dataset

Random Forest (CIC-IDS2017 RF)

Applying the Random Forest classifier to the CIC-IDS2017 dataset yielded outstanding results, with precision and recall values exceeding 85% for both classes. The F1 scores of 84% for benign traffic and 89% for malicious traffic indicate a well-balanced model capable of minimizing both false positives and false negatives. Random Forest's ensemble learning approach, which aggregates the outputs of multiple decision trees, likely contributed to its superior performance by reducing overfitting and improving generalization.

LSTM (CIC-IDS2017 LSTM)

The Long Short-Term Memory (LSTM) network, designed to capture temporal dependencies in sequential data, performed commendably on the CIC-IDS2017 dataset. Achieving F1-scores of 77% for benign and 82% for malicious traffic, the LSTM model effectively leveraged time-series patterns in the data. However, its performance slightly lagged behind the Random Forest model, potentially due to the complex nature of the dataset requiring more advanced feature extraction techniques.

Support Vector Machine (CIC-IDS2017 SVM)

The Support Vector Machine (SVM) model demonstrated strong classification capabilities with precision and recall values around 80% and 83%, respectively. The SVM's ability to find an optimal hyperplane for classification contributed to its robust performance, especially in distinguishing complex patterns in high-dimensional spaces. However, its computational efficiency and scalability may pose challenges when applied to larger datasets.

XGBoost (CIC-IDS2017 XGBoost)

XGBoost, an advanced gradient boosting algorithm, outperformed all other models, achieving precision and recall values close to 90%. The F1-scores of 86% for benign and 90% for malicious traffic underscore its exceptional predictive power and efficiency. XGBoost's ability to handle missing values, apply regularization, and optimize computational speed likely contributed to its superior performance in network intrusion detection.

Convolutional Neural Network (CIC-IDS2017 CNN)

The Convolutional Neural Network (CNN) model also performed well, with F1-scores of 81% for benign and 85% for malicious traffic. By capturing spatial patterns in the data, the CNN was able to identify subtle anomalies in network traffic. While not as effective as XGBoost or Random Forest, the CNN model's performance indicates its potential when combined with other techniques like feature extraction or hybrid models.

VII. Future Work

In addition to the specific roles mentioned in our game model (i.e. Godfather, Cultafia, Mafia, Detective, Doctor, Proof, and Townie), the Mafia game has also some other roles that we found their similarity with network components. Some of these roles and their equivalent in the network are as follows:

- **Alarmist:** Alarmist is a player in the Townie team, who can prevent the Cultafia from negotiating and recruiting the Townies. Every night, the Alarmist chooses a player, and if that player has been negotiated by the Mafia team, it will not join their team. Alarmist is equivalent to the security mechanisms that prevent the successful formation of the botnet army. MTD is one of these mechanisms that changes the attacking surface, and this change stops the spread of the bots.

- **Thief:** The Thief, who is one of Townie members, steals other players' votes. The Thief points to one or more people at night, and during the day, the votes of those people will be changed to be equal to the Thief, or in some cases their votes are ignored. This roles is equivalent to intrusion prevention mechanisms, which block the traffic from the malicious sources to specific destinations only for a specific time interval.
- **Sniper:** Sniper is one of the Townie players, who can shoot one of the Mafias in the night phase. Since the Sniper does not know who is in the Mafia team, its responsibility is to shot the players who are probably Mafia. This roles is also equivalent to the intrusion prevention mechanisms, that do not ever let the malicious node remain in the network, by blocking the packets that are from the malicious sources.
- **Governor:** The Governor can cancel the voting during the day, thus preventing anyone from being removed from the game. If the Governor cancels the vote on a bad day, this will help the Mafia team, because if the townies have a consensus on a mafia member, there will be no voting and that mafia will not leave the game. The role of Governor is equivalent to the security mechanisms in emergency conditions, when the malicious nodes are not detected, but a dangerous attack is threatening the network. So, all the traffics are blocked for a short time to disrupt the adversary. However, this blocking also disrupts the network functionality. But this is the only choice.
- **Reviver:** The Reviver can bring the removed players back into the game in the night phases. The number of revivals are limited and depends on the rules of the game. This role is equivalent to the passage of time for a host damaged by a DDoS attack. Such host will eventually return to its normal state in the network after a specific time.

As the future work, we have planned to also include the extra roles in our game model, and then to suggest their winning strategies. We can also evaluate the performance of a network which is modeled using our game model with these extra roles.

VIII. Discussions

This study introduces an innovative approach to network security by utilizing game-theoretic principles inspired by the Mafia game to enhance botnet detection. By establishing a mapping between roles in the Mafia game and cybersecurity adversaries, the research illustrates how botnet operators can be likened to the Mafia team, while network defenders represent Town members collaborating to identify and neutralize threats. The framework not only models adversarial deception and strategic interactions but also enhances traditional game-theoretic approaches by considering hidden roles and probabilistic interactions, thereby addressing fundamental challenges in detecting botnet controllers and compromised nodes.

Two distinct detection strategies emerge from the research: a linear relation strategy, which employs mathematically derived rules for effective suspicious node prioritization in large-scale networks, and a reinforcement learning-based strategy, which leverages machine learning to adjust detection policies dynamically. These strategies are rigorously evaluated through simulations, revealing their respective strengths and limitations in various network environments. Furthermore, the study demonstrates the practical applicability of these Mafia game-inspired detection mechanisms through real-world simulations using Mininet, offering valuable insights into the development of adaptive intrusion detection systems and cybersecurity frameworks that integrate deception-based modeling. Ultimately, this work effectively bridges the gap between theoretical game-theoretic modeling and practical applications in network defense, paving the way for future endeavors that combine reinforcement learning with rule-based decision-making for enhanced detection accuracy and scalability.

IX. Conclusion

In this paper, we have first presented the mathematical model of the Mafia game, and then applied this model on the networks. The researches on the Mafia game lack a general modeling of the game and also suggests weak strategies. Hence, we have considered almost all the game parameters in our model, and then suggested some strategies for each of the roles. Two different techniques, linear relation and reinforcement learning, are proposed for applying these strategies. For evaluating these strategies, we have extended the TPR and TNR metrics definitions to be matched with the Mafia game roles. Our suggested strategies for the Godfather, the Detective, and the Doctor are 15.5%, 11%, and 10% more accurate than the previously suggested strategies, respectively. Moreover, we have suggested strategies for the role of Cultafia, which is not considered by the existing researches.

We have also modeled a network that is compromised by the Mirai botnet with the proposed Mafia game model. Then we have applied the Detective's suggested strategies on this network to check its adaptability with the network. The emulation results obtained from Mininet show that the malicious nodes are detected with a TPR and TNR values of 71% and 91%, respectively, which is a satisfactory result.

X. Declaration of interests

- The authors (**Amir Javadpour Forough Ja'fari, Tarik Taleb, Chafika Benza¨ıd**) declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

- The authors (**Amir Javadpour Forough Ja'fari, Tarik Taleb, Chafika Benza¨ıd**) declare the following financial interests/personal relationships which may be considered as potential competing interests.

ACKNOWLEDGMENT

This research work is partially supported by the European Union's Horizon Europe research and innovation program HORIZON-JU-SNS-2022 under the RIGOUROUS project (Grant No. 101095933).

REFERENCES

[1] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A comprehensive survey on cyber deception techniques to improve honeypot performance," *Computers & Security*, p. 103792, 2024.
[2] A. Javadpour, P. Pinto, F. Ja'fari, and W. Zhang, "Dmaidps: A distributed multi-agent intrusion detection and prevention system for cloud iot environments," *Cluster Computing*, vol. 26, no. 1, pp. 367–384, 2023.
[3] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and B. Yang, "Scema: an sdn-oriented cost-effective edge-based mtd approach," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 667–682, 2022.
[4] A. Javadpour, F. Ja'fari, T. Taleb, Y. Zhao, B. Yang, and C. Benzaïd, "Encryption as a service for iot: Opportunities, challenges, and solutions," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7525–7558, 2023.
[5] A. Javadpour, F. Ja'fari, T. Taleb, H. Ahmadi, and C. Benzaïd, "Cybersecurity fusion: Leveraging mafia game tactics and reinforcement learning for botnet detection," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 6005–6011.
[6] MafiaScum, "Mafiascum wiki," https://wiki.mafiascum.net/, 2021, [Accessed: June 2022].
[7] F. Ja'fari, S. Mostafavi, K. Mizanian, and E. Jafari, "An intelligent botnet blocking approach in software defined networks using honeypots," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2993–3016, 2021.
[8] J.-H. Cho and N. Ben-Asher, "Cyber defense in breadth: Modeling and analysis of integrated defense systems," *The Journal of Defense Modeling and Simulation*, vol. 15, no. 2, pp. 147–160, 2018.
[9] D. Fraunholz, S. D. Anton, C. Lipps, D. Reti, D. Krohmer, F. Pohl, M. Tammen, and H. D. Schotten, "Demystifying deception technology: A survey," *arXiv preprint arXiv:1804.06196*, 2018.
[10] S. Lagraa, M. Husák, H. Seba, S. Vuppala, R. State, and M. Ouedraogo, "A review on graph-based approaches for network security monitoring and botnet detection," *International Journal of Information Security*, vol. 23, no. 1, pp. 119–140, 2024.
[11] W. Wu, J. Alvarez, C. Liu, and H.-M. Sun, "Bot detection using unsupervised machine learning," *Microsystem Technologies*, vol. 24, pp. 209–217, 2018.
[12] S. Haq and Y. Singh, "Botnet detection using machine learning," in *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE, 2018, pp. 240–245.
[13] R. H. Randhawa, N. Aslam, M. Alauthman, H. Rafiq, and F. Comeau, "Security hardening of botnet detectors using generative adversarial networks," *IEEE Access*, vol. 9, pp. 78 276–78 292, 2021.
[14] M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran, and I. Ashraf, "Hybrid machine learning model for efficient botnet attack detection in iot environment," *IEEE Access*, 2024.
[15] T. Al-Shurbaji, M. Anbar, S. Manickam, I. H. Hasbullah, N. ALfriehate, B. A. Alabsi, A. R. Alzighaibi, and H. Hashim, "Deep learning-based intrusion detection system for detecting iot botnet attacks: A review," *IEEE Access*, 2025.
[16] M. Albanese, S. Jajodia, and S. Venkatesan, "Defending from stealthy botnets using moving target defenses," *IEEE Security & Privacy*, vol. 16, no. 1, pp. 92–97, 2018.
[17] Y. Zhou, Q.-s. Li, Q. Miao, and K. Yim, "Dga-based botnet detection using dns traffic." *J. Internet Serv. Inf. Secur.*, vol. 3, no. 3/4, pp. 116–123, 2013.
[18] A. D. Kumar, H. Thodupunoori, R. Vinayakumar, K. Soman, P. Poornachandran, M. Alazab, and S. Venkatraman, "Enhanced domain generating algorithm detection based on deep neural networks," *Deep learning applications for cyber security*, pp. 151–173, 2019.
[19] T. Nguyen, M. P. Wellman, and S. Singh, "A stackelberg game model for botnet data exfiltration," in *Decision and Game Theory for Security: 8th International Conference, GameSec 2017, Vienna, Austria, October 23-25, 2017, Proceedings*. Springer, 2017, pp. 151–170.
[20] X. Chen, L. Xiao, W. Feng, N. Ge, and X. Wang, "Ddos defense for iot: A stackelberg game model-enabled collaborative framework," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9659–9674, 2021.
[21] A. Jakóbik, "Stackelberg game modeling of cloud security defending strategy in the case of information leaks and corruption," *Simulation Modelling Practice and Theory*, vol. 103, p. 102071, 2020.
[22] O. Tsemogne, Y. Hayel, C. Kamhoua, and G. Deugoué, "Game-theoretic modeling of cyber deception against epidemic botnets in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2678–2687, 2021.
[23] M. Braverman, O. Etesami, and E. Mossel, "Mafia: A theoretical study of players and coalitions in a partial information environment," *The Annals of Applied Probability*, vol. 18, no. 3, pp. 825–846, 2008.
[24] E. Yao, "A theoretical study of mafia games," *arXiv preprint arXiv:0804.0071*, 2008.
[25] P. Migdał, "A mathematical model of the mafia game," *arXiv preprint arXiv:1009.1031*, 2013.

[26] J. Xu, P. Paliyawan, R. Thawonmas, and T. Harada, "Player dominance adjustment: Promoting self-efficacy and experience of game players by adjusting dominant power," in *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2019, pp. 487–488.
[27] S. Xiong, W. Li, X. Mao, and H. Iida, "Mafia game setting research using game refinement measurement," in *International Conference on Advances in Computer Entertainment*. Springer, 2017, pp. 830–846.
[28] H. Ri, X. Kang, M. N. A. Khalid, and H. Iida, "The dynamics of minority versus majority behaviors: A case study of the mafia game," *Information*, vol. 13, no. 3, p. 134, 2022.
[29] L. Zhou and Y.-w. Sung, "Cues to deception in online chinese groups," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. IEEE, 2008, pp. 146–146.
[30] S. Demyanov, J. Bailey, K. Ramamohanarao, and C. Leckie, "Detection of deception in the mafia party game," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015, pp. 335–342.
[31] D. Katagami, S. Takaku, M. Inaba, H. Osawa, K. Shinoda, J. Nishino, and F. Toriumi, "Investigation of the effects of nonverbal information on werewolf," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2014, pp. 982–987.
[32] C. Girlea, R. Girju, and E. Amir, "Psycholinguistic features for deceptive role detection in werewolf." in *HLT-NAACL*, 2016, pp. 417–422.
[33] S. Nagayama, J. Abe, K. Oya, K. Sakamoto, H. Shibuki, T. Mori, and N. Kando, "Strategies for an autonomous agent playing the "werewolf game" as a stealth werewolf," in *Proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial2019)*, 2019, pp. 20–24.
[34] D. Tellols, "Are talkative ai agents more likely to win the werewolf game?" in *Proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial2019)*, 2019, pp. 11–14.
[35] D. Katagami, M. Kanazawa, F. Toriumi, H. Osawa, M. Inaba, and K. Shinoda, "Movement design of a life-like agent for the werewolf game," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2015, pp. 1–7.
[36] Y. Hirata, M. Inaba, K. Takahashi, F. Toriumi, H. Osawa, D. Katagami, and K. Shinoda, "Werewolf game modeling using action probabilities based on play log analysis," in *International Conference on Computers and Games*. Springer, 2016, pp. 103–114.
[37] Y. Kano, C. Aranha, M. Inaba, F. Toriumi, H. Osawa, D. Katagami, T. Otsuki, I. Tsunoda, S. Nagayama, D. Tellols *et al.*, "Overview of aiwolfdial 2019 shared task: Contest of automatic dialog agents to play the werewolf game through conversations," in *Proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial2019)*, 2019, pp. 1–6.
[38] S. Ibraheem, G. Zhou, and J. DeNero, "Putting the con in context: Identifying deceptive actors in the game of mafia," in *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.
[39] M. Khan and C. Aranha, "A novel weighted ensemble learning based agent for the werewolf game," *arXiv preprint arXiv:2205.09813*, 2022.
[40] C. L. Gîrlea, E. Amir, and R. Girju, "Tracking beliefs and intentions in the werewolf game," in *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2014.
[41] G. Halim, "Strategies in the game of mafia," *Horizons*, vol. 6, pp. 135–139, 2021.
[42] X. Bi and T. Tanaka, "Human-side strategies in the werewolf game against the stealth werewolf strategy," in *International Conference on Computers and Games*. Springer, 2016, pp. 93–102.
[43] Y. Hatori, S. Wu, Y. Lin, and T. Utsuro, "Mining preferences on identifying werewolf players from werewolf game logs," in *International Conference on Entertainment Computing*. Springer, 2017, pp. 353–356.
[44] Y. Lin, M. Baba, and T. Utsuro, "Generating a werewolf game log digest of inferring each player's role," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, 2016, pp. 1–6.
[45] Y. Lin, M. Kasamatsu, T. Chen, T. Fujita, H. Deng, and T. Utsuro, "Automatic annotation of werewolf game corpus with players revealing oneselves as seer/medium and divination/medium results," in *Workshop on Games and Natural Language Processing*, 2020, pp. 85–93.
[46] N. Nide and S. Takata, "Tracing werewolf game by using extended bdi model," *IEICE TRANSACTIONS on Information and Systems*, vol. 100, no. 12, pp. 2888–2896, 2017.
[47] A. Azaria, A. Richardson, and S. Kraus, "An agent for deception detection in discussion based environments," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015, pp. 218–227.
[48] N. Nakamura, M. Inaba, K. Takahashi, F. Toriumi, H. Osawa, D. Katagami, and K. Shinoda, "Constructing a human-like agent for the werewolf game using a psychological model based multiple perspectives," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–8.
[49] M. Kondoh, K. Matsumoto, and N. Mori, "Development of agent predicting werewolf with deep learning," in *International Symposium on Distributed Computing and Artificial Intelligence*. Springer, 2018, pp. 18–26.
[50] M. Hagiwara, A. Moustafa, and T. Ito, "Using q-learning and estimation of role in werewolf game," in *Proceedings of the Annual Conference of JSAI 33rd (2019)*. The Japanese Society for Artificial Intelligence, 2019, pp. 2O5E303–2O5E303.

[51] T. Wang and T. Kaneko, "Application of deep reinforcement learning in werewolf game agents," in *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, 2018, pp. 28–33.
[52] M. Eger and C. Martens, "A study of ai agent commitment in one night ultimate werewolf with human players," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 15, no. 1, 2019, pp. 139–145.
[53] Q. Chang, "A simulator for analyzing the balance of mafia game," in *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*. IEEE, 2020, pp. 622–627.
[54] F. Toriumi, H. Osawa, M. Inaba, D. Katagami, K. Shinoda, and H. Matsubara, "Ai wolf contest—development of game ai using collective intelligence—," in *Computer Games*. Springer, 2016, pp. 101–115.
[55] B. Wang, H. Osawa, T. Toyono, F. Toriumi, and D. Katagami, "Development of real-world agent system for werewolf game." in *AAMAS*, 2018, pp. 1838–1840.
[56] M. Plus, "Mafia plus," https://www.youtube.com/channel/UCzZJv5EsGb-h1HDEPa2o4Bg/videos, 2020, [Accessed: June 2022].
[57] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093–1110.
[58] A. Javadpour, F. Ja'fari, T. Taleb, and C. Benzaïd, "Enhancing 5g network slicing: Slice isolation via actor-critic reinforcement learning with optimal graph features," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 31–37.
[59] ——, "5g slice mutation to overcome distributed denial of service attacks using reinforcement learning," in *2024 17th International Conference on Security of Information and Networks (SIN)*. IEEE, 2024, pp. 1–9.

APPENDIX1: DATASET DESCRIPTION

The datasets "http://mosaic-lab.org/data-set/mafia-dataset.zip" used in this study are derived from the Iranian TV Mafia show (ITM), broadcast on the Salamat channel. The details of each dataset, corresponding to different seasons and rounds, are presented below.

TABLE V
DATASET DESCRIPTIONS

Dataset	Program	Description
ITM-8-s1m	Men Semi-final – first round	Dataset from Season 8, capturing the first semi-final round for men.
ITM-8-s2m	Men Semi-final – second round	Dataset from Season 8, covering the second semi-final round for men.
ITM-9-p1w	Women preliminary – first round	Dataset from Season 9, detailing the first preliminary round for women.
ITM-9-p2w	Women preliminary – second round	Dataset from Season 9, describing the second preliminary round for women.
ITM-9-fw	Women Final	Dataset from Season 9, documenting the final round for women.
ITM-9-q5m	Mixed Quarter-final	Dataset from Season 9, focusing on the quarter-final round with mixed participants.

SHOW AND PROGRAM DESCRIPTIONS

TABLE VI
SHOW AND PROGRAM DESCRIPTIONS

Show	Show Description	Season	Program	Program Description
ITM (Iran TV Mafia)	Iran TV show from Salamat channel (mosabegheye mafiaye salamat)	8	s1m	Men Semi-final – first round
		8	s2m	Men Semi-final – second round
		9	p1w	Women preliminary – first round
		9	p2w	Women preliminary – second round
		9	fw	Women Final
		9	q5m	Mixed Quarter-final

SAMPLE DATASET PREVIEWS

TABLE VII
SAMPLE OF ITM-9-FW DATASET INCLUDING NAN VALUES

max width=

Turn	Turn Info 1	Challenge	Challenge Info 1	Court	Court Info 1	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	1	2	3	4	5	6	7	8	9	10	NaN	NaN
Roles	Reporter	Townie	Godfather	Cultafia	Sniper	Doctor	Detective	Proof	Mafia	Townie	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

TABLE VIII
SAMPLE OF ITM-9-P1W DATASET INCLUDING NAN VALUES

max width=

Winner	Townie	Unnamed: 2	Turn	Turn Info 2	Challenge	Challenge Info 2	Court	Court Info 2	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	1	2	3	4	5	6	7	8	9	10	NaN	NaN	NaN
Roles	Proof	Sniper	Mafia	Townie	Doctor	Cultafia	Reporter	Detective	Townie	Godfather	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

TABLE IX
SAMPLE OF ITM-9-P2W DATASET INCLUDING NAN VALUES

max width=

Winner	Townie	Unnamed: 2	Turn	Turn Info 3	Challenge	Challenge Info 3	Court	Court Info 3	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	1	2	3	4	5	6	7	8	9	10	NaN	NaN	NaN
Roles	Cultafia	Detective	Reporter	Proof	Godfather	Sniper	Townie	Doctor	Mafia	Townie	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

APPENDIX 2: DATASET DESCRIPTION

This appendix provides a detailed description of the CIC-IDS2017 dataset, used for evaluating intrusion detection systems. Each dataset corresponds to a different type of network traffic, categorized by attack type and normal activity.

TABLE X
DATASET DESCRIPTIONS

Dataset	Program	Description
Benign-Monday	Normal Traffic	Captures legitimate network traffic without any malicious activity (Monday).
Botnet-Friday	Botnet Attack	Contains network traffic with botnet-related activities, including command and control traffic (Friday).
Bruteforce-Tuesday	Bruteforce Attack	Traffic logs capturing multiple unauthorized login attempts, simulating credential stuffing attacks (Tuesday).
DDoS-Friday	DDoS Attack	Traffic showing Distributed Denial of Service (DDoS) attack patterns with high packet rates (Friday).
DoS-Wednesday	DoS Attack	Denial of Service attack events targeting specific network resources to cause disruption (Wednesday).
Infiltration-Thursday	Infiltration Attack	Simulated unauthorized internal network access mimicking an advanced persistent threat (Thursday).
Portscan-Friday	Portscan Attack	Large-scale port scanning attempts to detect open services and vulnerabilities (Friday).
WebAttacks-Thursday	Web Attack	Web-based attacks, including SQL injection, cross-site scripting (XSS), and command injection (Thursday).

SHOW AND PROGRAM DESCRIPTIONS

The following table outlines the network activity captured in the dataset, classified by attack category and normal behavior.

TABLE XI
SHOW AND PROGRAM DESCRIPTIONS

Show	Show Description	Season	Program	Program Description
CIC-IDS2017	Intrusion Detection Evaluation Dataset	2017	Benign	Legitimate network traffic without anomalies.
		2017	Botnet	Simulated botnet traffic, including C2 communication.
		2017	Bruteforce	Repeated unauthorized login attempts to mimic account takeover attempts.
		2017	DDoS	High-volume traffic floods targeting network availability.
		2017	DoS	Single-source attack flooding specific network services.
		2017	Infiltration	Exploit-based unauthorized access attempts from an external entity.
		2017	Portscan	Active probing of network ports to identify open services.
		2017	Web Attacks	Web-based exploitation techniques targeting application vulnerabilities.

SAMPLE DATASET PREVIEWS

The following table presents a subset of the network traffic records, including timestamps, IP addresses, protocol types, and attack labels.

TABLE XII
SAMPLE OF NETWORK TRAFFIC DATASET

Timestamp	Source IP	Destination IP	Protocol	Bytes Sent	Bytes Received	Attack Type
2023-01-01 00:00:00	192.168.1.3	10.0.0.1	ICMP	869	6520	DDoS
2023-01-01 00:01:00	192.168.1.1	10.0.0.3	ICMP	7049	1284	Normal
2023-01-01 00:02:00	192.168.1.3	10.0.0.2	TCP	2533	4655	Normal
2023-01-01 00:03:00	192.168.1.3	10.0.0.3	ICMP	5411	3485	Bruteforce
2023-01-01 00:04:00	192.168.1.1	10.0.0.3	UDP	5151	6496	Bruteforce

APPENDIX 3: MAFIA GAME INTERACTION GRAPHS

Day 1

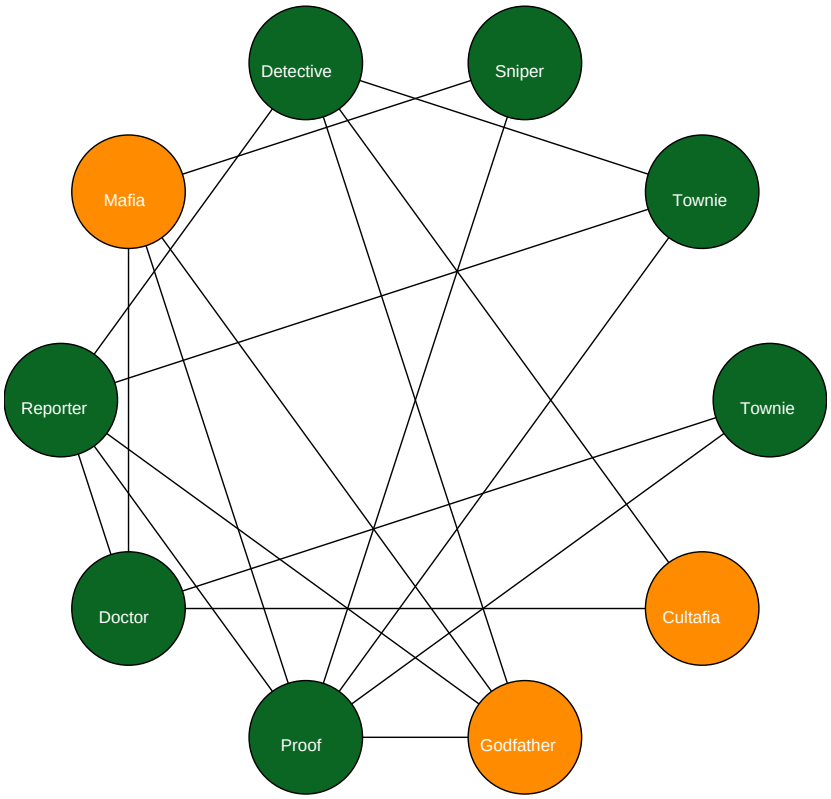


Fig. 20. Interaction graph for Day 1

The graph for Day 1 illustrates the initial interaction patterns among the players. Mafia members like the **Godfather**, **Cultafia**, and **Mafia** are marked in orange, while the rest of the players, representing town roles, are shown in green. The connections between players are represented by lines whose thickness reflects the frequency of interactions or votes. On Day 1, the connections tend to be sparse and thin, indicating cautious interactions as players are still gathering information.

Day 2

Day 2's graph starts to show more defined patterns of alliances and suspicions. The **Mafia members** might attempt to subtly coordinate, reflected in slightly thicker lines between them. Town players could begin forming alliances, leading to clusters of thicker green connections. The dynamics start to shift as players gain more information and make more strategic decisions.

Day 3

By Day 3, the tension increases, and this is evident in the graph through denser connections. You may observe more prominent links between players suspected of being Mafia. Thicker lines connecting orange nodes (Mafia roles) may indicate increased communication or coordinated efforts, while town members might target these players, leading to complex connection patterns.

Day 4

Day 4 often marks a turning point in the game. The graph may show heavy clustering around certain players, indicating focused suspicion or alliance-building. If a **Mafia** member has been exposed, you'll see a decrease in connections for that player, or conversely, very thick lines if they are heavily targeted. Surviving **Mafia** might show more covert connections, while **town roles** form visible alliances.

Day 5

The network on Day 5 is likely highly polarized. Clear factions emerge, with thick lines showing strong trust or repeated accusations. If the **Mafia** still have numbers, you might see interconnected orange nodes, indicating their persistent coordination. The **town's** efforts will be concentrated, with clusters of green nodes reflecting their attempts to eliminate remaining threats.

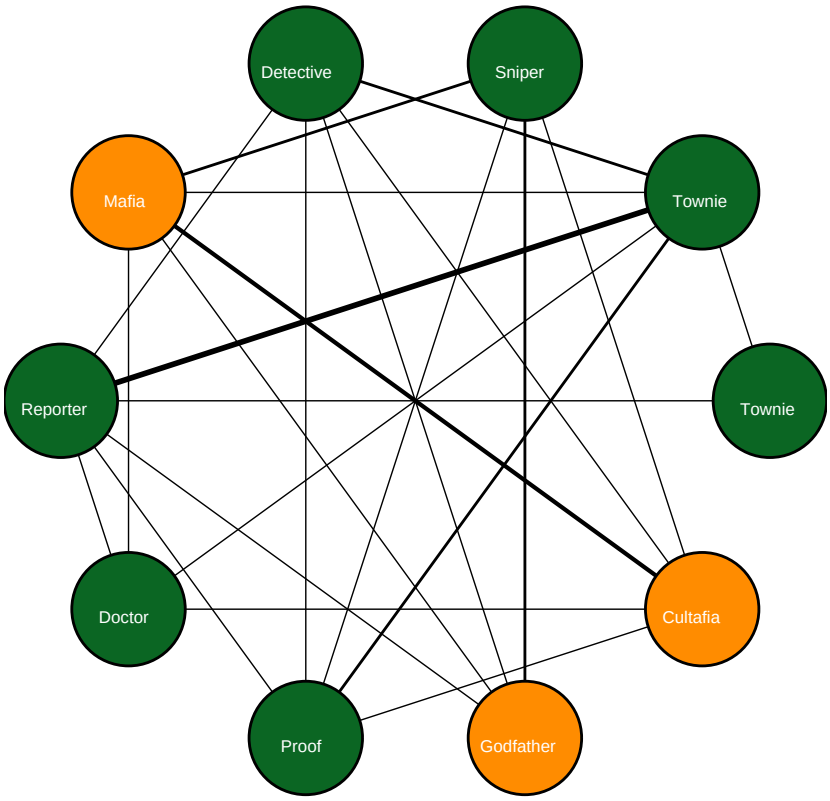


Fig. 21. Interaction graph for Day 2

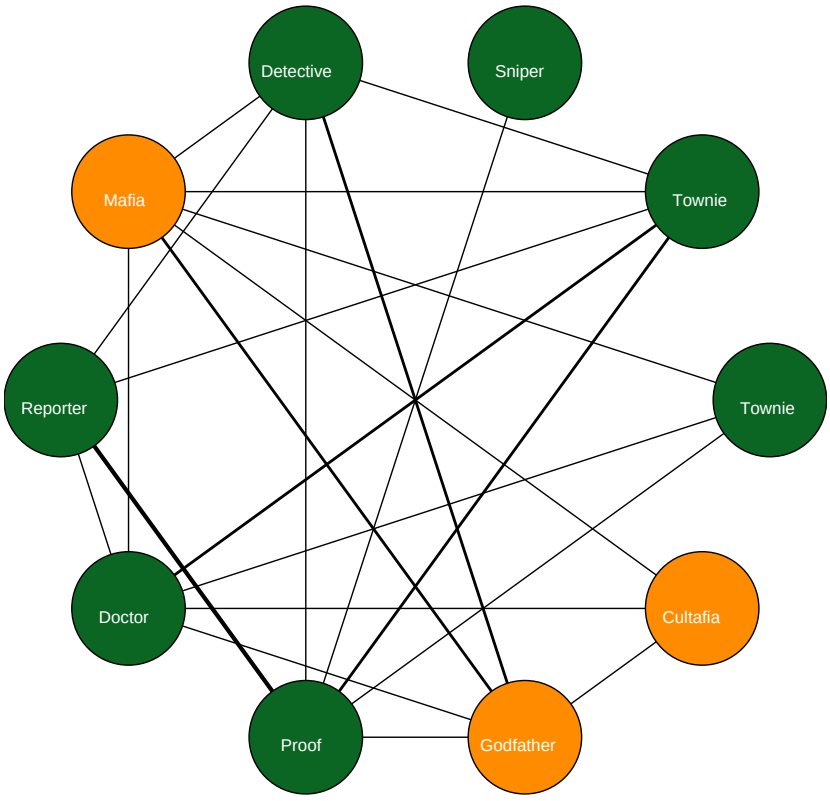


Fig. 22. Interaction graph for Day 3

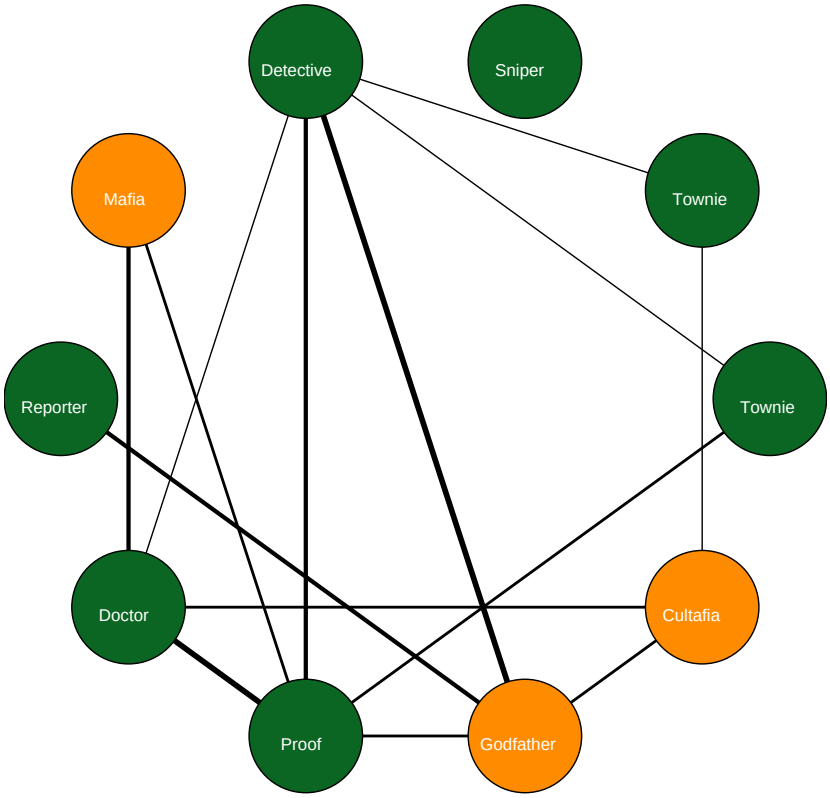


Fig. 23. Interaction graph for Day 4

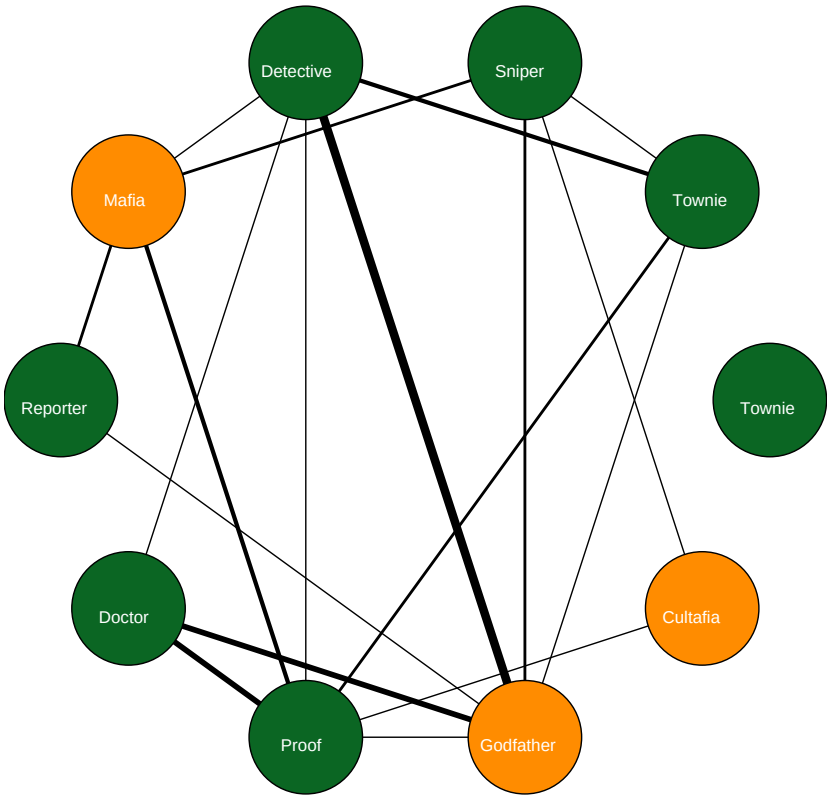


Fig. 24. Interaction graph for Day 5

Day 6

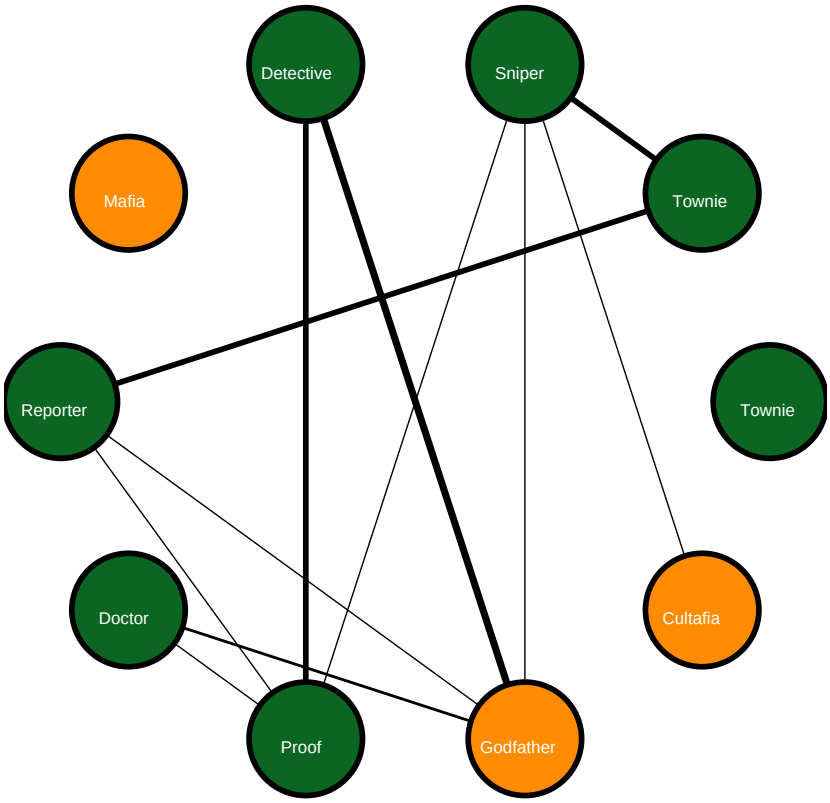


Fig. 25. Interaction graph for Day 6

By Day 6, the graph typically becomes simpler but more intense. There may be fewer players, but the connections are strong and direct, reflecting critical decision-making. If **Mafia** members remain, their connections might appear isolated or sparsely linked to others. The **town's** focus will be sharp, with intense interactions between the final suspects and allies.

Amir Javadpour holds a Ph.D. in Mathematics/Cybersecurity from Guangzhou University, China. His academic journey is distinguished by numerous publications in highly-ranked journals and prestigious conferences. These works span a diverse range of topics, reflecting his deep expertise in Cybersecurity, Cloud Computing, Software-Defined Networking (SDN), etc. Additionally, he serves as a dedicated Technical Program Committee (TPC) member for several international conferences. Dr. Javadpour actively collaborates internationally, particularly with European consortiums on funded projects such as Inspire-5Gplus (https://www.inspire-5gplus.eu/) and Rigourous (https://rigourous.eu/). These partnerships have resulted in significant contributions to the field, with his work being featured in top-tier journals and conferences, including Globecom, IEEE Transactions on Industrial Informatics (TII), IEEE Transactions on Information Forensics and Security (TIFS), IEEE Transactions on Network and Service Management (TNSM), and ACM Transactions on Sensor Networks (TOSN). In addition to his research and publication efforts, he is deeply committed to mentoring and supervising Master's and Doctoral students. His extensive experience in this area has equipped him with the skills and confidence necessary to lead a research group and conduct independent, high-impact research.

Forough Ja'fari is a Senior Researcher in cybersecurity and computer science. She received her Bachelor's degree from Sharif University of Technology and her Master's degree in Computer Network Engineering from Yazd University, Iran. She is a visiting scholar researcher at Guangzhou University, China. Software-Defined Networking (SDN), Intrusion Detection Systems (IDS), Internet of Things (IoT), Moving Target Defence (MTD), and Machine Learning are some of her research interests. She is currently a Guest Editor (GE) of Cluster Computing (CLUS) Journal, as well as a reviewer for several journals and conferences.

Tarik Taleb Prof. Tarik Taleb is currently a Chair Professor at Ruhr University Bochum, Bochum, Germany. Prior to that, he was a full professor at the Centre for Wireless Communications (CWC) – Networks and Systems Unit, Faculty of Information Technology and Electrical Engineering, The University of Oulu. Between Oct. 2014 and Dec. 2021, he was a Professor at the School of Electrical Engineering, Aalto University, Finland. Prior to that, he was working as Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd, Heidelberg, Germany. He was then leading the NEC Europe Labs Team working on R&D projects on carrier cloud platforms. Before joining NEC and till Mar. 2009, he worked as assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI, the second largest mobile operator in Japan. From Oct. 2005 till Mar. 2006, he worked as a research fellow at the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B. E degree in Information Engineering with distinction, M.Sc. and Ph.D. degrees in Information Sciences from Tohoku Univ., in 2001, 2003, and 2005, respectively. Prof. Taleb's research interests lie in the field of telco cloud, network softwarization and network slicing, AI-based software defined security, immersive communications, mobile multimedia streaming, next generation mobile networking. Prof. Taleb was also directly engaged in the development and standardization of the Evolved Packet System as a member of 3GPP's System Architecture working group 2. Prof. Taleb served on the IEEE Communications Society Standardization Program Development Board. As an attempt to bridge the gap between academia and industry, Prof. Taleb founded the "IEEE Workshop on Telecommunications Standards: from Research to Standards", a successful event that got awarded the "best workshop award" by IEEE Communication Society (ComSoC). Based on the success of this workshop, Prof. Taleb also founded and served as the steering committee chair of the IEEE Conf. on Standards for Communications and Networking. Prof. Taleb served as the general chair of the 2019 edition of the IEEE Wireless Communications and Networking Conference (WCNC'19) held in Marrakech, Morocco. He was the guest editor in chief of the IEEE JSAC Series on Network Softwarization and Enablers. He was on the editorial board of the IEEE Transactions on Wireless Communications, IEEE Wireless Communications Magazine, IEEE Journal on Internet of Things, IEEE Transactions on Vehicular Technology, IEEE Communications Surveys & Tutorials, and a number of Wiley journals. Till Dec. 2016, he served as chair of the Wireless Communications Technical Committee, the largest in IEEE ComSoC. He also served as Vice Chair of the Satellite and Space Communications Technical Committee of IEEE ComSoc (2006 - 2010). Prof. Taleb is the recipient of the 2021 IEEE ComSoc Wireless Communications Technical Committee Recognition Award (Dec. 2021), the 2017 IEEE ComSoc Communications Software Technical Achievement Award (Dec. 2017) for his outstanding contributions to network softwarization. He is also the (co-) recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize (May 2017), the 2009 IEEE ComSoc Asia-Pacific Best Young Researcher award (Jun. 2009), the 2008 TELECOM System Technology Award from the Telecommunications Advancement Foundation (Mar. 2008), the 2007 Funai Foundation Science Promotion Award (Apr. 2007), the 2006 IEEE Computer Society Japan Chapter Young Author Award (Dec. 2006), the Niwa Yasujirou Memorial Award (Feb. 2005), and the Young Researcher's Encouragement Award from the Japan chapter of the IEEE Vehicular Technology Society (VTS) (Oct. 2003).

Chafika Benzaïd is currently a senior research fellow at University of Oulu, Finland. Between Nov. 2018 and Dec. 2021, she was senior researcher at Aalto University. Before that, she worked as an associate professor at University of Sciences and Technology Houari Boumediene (USTHB). She holds Engineer, Magister and "Doctorat ès Sciences" degrees from USTHB. Her research interests lie in the field of 5G/6G, SDN, Network Security, AI Security, and AI/ML for zero-touch security management. She is an ACM professional member.