

Multi-agent Reinforcement Learning-based Passenger Spoofing Attack on Mobility-as-a-Service

Kai-Fung Chu, *Member, IEEE*, Weisi Guo, *Senior Member, IEEE*

Abstract—Cyber-physical systems, such as smart transportation, face security threats from both digital and physical realms. Recently, Mobility-as-a-Service (MaaS) has emerged as a novel transportation concept, offering passengers access to diverse mobility services via a unified platform. Central to this system is the smart MaaS coordinator, tasked with tailoring services to passengers based on their profiles and behaviors. However, the coordination of heterogeneous passengers introduces vulnerabilities, enabling malicious entities to exploit the system by impersonating priority passengers with falsified information. Effective detection mechanisms require a deep understanding of the spoofing process. This paper investigates threats to the smart MaaS coordinator, unveiling a new reinforcement learning-based attack named the passenger spoofing attack, which aims to mitigate the risk of inadvertently exposing MaaS vulnerabilities post-deployment. This attack leverages feedback from actions and experiences to manipulate system profitability and passenger satisfaction by generating false passenger information. Furthermore, our research reveals that multi-agent reinforcement learning, accounting for spatial distribution among malicious agents and passengers, strengthens the attack. Through simulations based on datasets from New York City and synthetic sources, we demonstrate that the attack can significantly diminish profit and passenger satisfaction, reaching levels as low as 70% and 50%, respectively. Spatial analysis indicates an effective distance of approximately two nodes from the origin or destination. This study enriches our comprehension of the vulnerabilities inherent in smart coordinators within MaaS, enabling the development of robust countermeasures against malicious actors.

Index Terms—Passenger spoofing attack, multi-agent reinforcement learning, multimodal transport, intelligent transportation systems.



1 INTRODUCTION

Mobility-as-a-Service (MaaS), as a cyber-physical transportation system unifying different transport modes, provides comprehensive transport-related services, including journey planning, reservation, and bundle purchasing across multiple mobility providers [1]. The integration of mobility providers forms an extensive transport network that provides passengers with extra transport choices and flexibility. In the integrated multimodal transport network of MaaS, the services usually begin with satisfactory journey planning based on the passengers' origin and destination. Passengers may enjoy the benefits of a well-designed journey planner such as discounted bundles and traffic congestion avoidance. Although an optimal journey can be obtained by solving these problems, the preference of individual passengers has not been considered in the models. In other words, the solutions are optimal with respect to the predefined objective, but not the best journey for all passengers with the same origin and destination. Therefore, the authors in [2] use the artificial intelligence (AI) approach to incorporate passenger behavior into the journey planning

process, which increases passenger satisfaction, retention rate, and profit.

While the advancement of MaaS is appealing to meet the increasing demand for multi-modal transportation and optimizing public transport infrastructure. The transition to a MaaS ecosystem involves integrating data and services into a single platform, which also expands the cyber attack surface for MaaS operators and consumers [3]. Despite extensive research on the benefits of MaaS, there has been relatively little focus on the cyber security and privacy risks associated with the MaaS ecosystem.

A survey [4] has summarized the MaaS architecture and the corresponding insider threats of each component. Multiple insiders such as developers, service administrators, managers, etc., may perform various attacks on MaaS. Those attacks may cause serious consequences such as information leakage, operational failure, and economic loss. People's trust in these new systems and technologies will be diminished due to security loopholes and accidents. For example, the authors in [5] studied the optimal attack method based on Denial-of-Service (DoS) attacks and the corresponding resiliency of MaaS. The authors concluded that the DoS attacks cost more than \$ 15 US dollars per unit for protection. Another reference [6] investigated the information processing mechanism by the MaaS journey planner and discovered that there is a loophole in passenger information leakage in the AI model. Identifying the possible attack methods and mitigating those risks can enhance societal acceptance and adaptation of the system and AI technologies.

AI is one of the popular and state-of-the-art technolo-

This work was supported by EPSRC MACRO - Mobility as a service: Managing Cybersecurity Risks across Consumers, Organisations and Sectors (EP/V039164/1)

Kai-Fung Chu is with the School of Aerospace, Transport and Manufacturing, Cranfield University, Milton Keynes, MK43 0AL, UK, and the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK. (e-mail: kfc35@cam.ac.uk)

Weisi Guo is with the School of Aerospace, Transport and Manufacturing, Cranfield University, Milton Keynes, MK43 0AL, UK. (e-mail: weisi.guo@cranfield.ac.uk)

gies incorporated into MaaS and intelligent transportation systems due to its success as in [7], [2], and [8]. However, AI is known to be a black-box method and we are still far from having a fully explainable AI [9]. Without the detailed causality on the decision made by the AI, it is challenging to guarantee that the decisions are uncorrupted since there are AI-focused attack methods such as data poisoning attacks [10]. For the MaaS that uses AI for planning journeys based on the heterogeneity of passengers as in [2], a malicious agent may take advantage of this heterogeneity by acting as a passenger with falsifying information that is likely to be prioritized. This may increase passenger travel time and cost and decrease passenger satisfaction with mobility services. A pilot study [11] has shown the potential threat to the MaaS. Therefore, we must re-evaluate the security of the MaaS system, especially for those using AI technology.

In this paper, we investigate the threat in the smart coordinator and identify a new reinforcement learning-based attack, named passenger spoofing attack (PSA), which leverages the diversity of passengers and the heterogeneous service planning to prioritize malicious agents in the journey planning process. The malicious agent learns to generate falsifying characteristics and satisfaction to be queried for service based on the current state of other passengers. The falsifying information is generated by a neural network trained by a reinforcement learning algorithm that uses feedback from actions and experiences. For multi-agent scenarios, we discover that a spatial discount factor that scales the reward of other agents based on the distance can be used to strengthen the attack if the parameter is set properly. We simulate two scenarios based on New York City and a synthetic dataset to evaluate the attack's influence on profit and passenger satisfaction. We find that the attack can reduce about 70% of profit and reduce about 50% of passenger satisfaction to the lowest level. From the spatial analysis, we find that the effective distance of the attack is about two nodes from the origin or destination. The main contributions of this paper can be summarized as follows:

1. A security threat of a smart coordinator in the MaaS system, that utilizes the diversity of passengers and the heterogeneous service planning in AI, is investigated.
2. We identify a new attack based on reinforcement learning, named PSA, to the smart coordinator that can prioritize malicious agents in the journey planning process and lower the satisfaction of other regular passengers and system profit.
3. For the scenario with multiple malicious agents, we discover that incorporating multi-agent interactions via the spatial distribution among the malicious agents and passengers can be used to strengthen the attack if the parameter is set properly.
4. Experiments on two simulated scenarios based on real-world and synthetic datasets were conducted to study the attack influence on passenger satisfaction and profit, as well as the sensitivity of malicious agent proportion and spatial distance. The results are useful for developing corresponding detection and defense mechanism.

This paper is of great importance as it delves into a

novel and critical cybersecurity challenge in the realm of MaaS systems. By investigating the PSA, the research not only identifies a previously unexplored threat but also underscores the potential consequences of such an attack. The findings shed light on how the PSA can significantly diminish profit and passenger satisfaction within MaaS platforms, emphasizing the urgency of addressing this security concern. Furthermore, the sophisticated methodology employed, combining multi-agent reinforcement learning and spatial discount factors, offers valuable insights into how attackers can manipulate the system. These insights extend beyond MaaS and can be applied to enhance security in various smart systems. As a result, the paper serves as a crucial resource for policymakers, MaaS providers, and researchers seeking to bolster security measures and countermeasures in the face of emerging cybersecurity threats.

The rest of this paper is organized as follows. Section 2 reviews the related works of MaaS and the security threats. Section 3 illustrates the MaaS model and journey planning problem. Section 4 presents the reinforcement learning-based PSA. Section 5 simulates the attacks in two simulated MaaS scenarios and analyzes the attack influence on the system. Finally, the paper is concluded in Section 6.

2 RELATED WORK

2.1 MaaS Ecosystem and Journey Planning

Mobility-as-a-Service (MaaS) is an emerging and dynamic mobility paradigm that integrates various transportation services. According to [12], the business ecosystem of MaaS encompasses a network of interconnected firms that exert influence on the value creation and capture processes of the focal firm, namely the MaaS provider. This business ecosystem, as stated by Kamargianni and Matyas [12], consists of multiple layers. At its core, the ecosystem encompasses the MaaS provider. Extending beyond the core, the ecosystem incorporates the extended enterprise layer, which broadens the perspective to encompass complementors and second-layer suppliers within the business supply chain. Finally, the outermost layer of the ecosystem comprises entities such as regulators, unions, universities, research bodies, investors, and stakeholders, thus further enriching the business ecosystem.

Given its composite nature involving multiple transport operators, ensuring the provision of optimal routes across the multimodal transport network is of paramount importance in MaaS research. Effective journey planning plays a pivotal role in coordinating the resources of mobility providers to meet passenger demand. Consequently, the literature contains numerous studies addressing the optimization of journey planning in multimodal transport networks, specifically tailored to the unique requirements of MaaS. Dib et al. proposed a novel heuristic approach, combining Genetic Algorithm with Variable Neighborhood Search, to solve the network scalability issue of a multicriteria journey planning problem in multimodal transport networks. Their method outperforms traditional algorithms in terms of computational efficiency and solution quality, particularly in real-world routing scenarios in the Paris metropolitan area [13]. Wang et al. proposed a mathematical framework for designing transit-oriented multimodal transportation systems

that minimize the total cost of mobility providers and passengers with considering travelers' choices including travelers' behavior and estimating agencies' costs. Using the proposed algorithm, they develop a global search framework for optimal design. The study reveals conditions where transportation modes are excluded, providing insights into system behavior and cost implications. Numerical examples validate the model, highlighting the role of different modes and factors like subways and autonomous vehicles [14].

2.2 MaaS Security Threats

As a cyber-physical system, MaaS faces security threats at data and physical levels. Callegati et al. summarized possible attacks related to insider activities, including data leakage, crowdsourcing attacks and data manipulation, and device misbehavior. For data leakage, malicious agents might obtain information by manipulating sensors to check the presence of people. For crowdsourcing attacks and data manipulation, fabricated data may be injected into the system to change the service behavior. For device misbehavior, the hardware and software vulnerability can be exploited to cause a system failure. They extend previous work by proposing countermeasures to mitigate these threats and address the emerging Cloud-of-Things for mobility. A novel overlay networking architecture based on gossip protocols is introduced to limit insider attacks, allowing users to share mobility information while constraining data access by insiders and optimizing request routing [4].

The next-generation connected and autonomous vehicles (CAV), as a core component in the intelligent transportation systems and MaaS, could be vulnerable to malicious attacks. [15] reviewed various attack and defense strategies on CAV communication and sensor data. To name a few, attacks on communication includes spoofing attack, man-in-the-middle attacks, jamming attack, masquerade attack, black-hole attack, replay attack, bus-off attacks, denial of service (DoS) attacks, injection attacks, timing attacks, impersonation attacks, slight attack, spamming attack, and tunneling attack. For attacks on sensor data, malicious agents can produce incorrect messages, obstruct sensor data, and intervene in autonomous driving by hacking into the CAV computing device. Tsao et al. suggested that sensitive mobility data may leak during communication between mobility providers and municipal authorities. Therefore, they proposed an interactive protocol using the commitment and zero-knowledge proofs to allow municipal authorities to obtain insights from mobility data without revealing the trade secrets or sensitive data of mobility providers' users. It can be extended to scenarios with multiple mobility providers via secure multiparty computation [16].

2.3 AI Security Threats

AI is known to be vulnerable to adversarial attacks [17] despite its extraordinary performance. Many attacks can be performed against AI systems. Data poisoning attacks [10] cause the model to underperform by injecting malicious samples into the training dataset. Evasion Attacks [18] is used to evade a deployed AI system by carefully manipulating attack samples. Adversarial examples are generated by purposely designed methods that cause mistakes in the

model output. Fast gradient sign method [19] is used to generate a small perturbation, which is added to the image to cause misclassification. The perturbation is equal to the sign of the elements of the gradient of the cost function that is sufficiently small and hard to be noticed by humans. [20] shows that many state-of-the-art defense mechanisms cannot detect those adversarial examples generated by the attack methods proposed in [21].

On the other hand, AI may suffer from backdoor attacks. An invisible backdoor with covert and scattered triggers can be embedded into deep neural networks as presented in [22]. The backdoor attack is not only limited to a single target against a single trigger backdoor, but also multi-target against a single trigger backdoor, and a single target against multi-trigger backdoors [23]. The backdoor attack can apply to transfer learning tasks where commonly adopted defenses can be defeated [24]. Even worse, the attack can be extended to the training phase in addition to the inference phase [25]. Therefore, the backdoor attacks attract much attention, which requires further studies. Furthermore, the deployed AI model is usually trained by many training samples containing sensitive and valuable information to malicious agents. Model extraction attacks can reconstruct the AI model or training sample information for malicious uses. For example, membership inference attacks [26] can determine whether a data sample was in the training dataset. Model inversion attacks [27] tried to infer the data samples used to train the AI model. Property inference attacks [28] focus on the inference of global properties of the training dataset.

2.4 Attacks on Transportation Systems

There are considerable studies on attacks on CAVs and transportation systems [29]. The adversarial examples can hardly be detected and may cause misbehavior and under-performance of CAVs and transportation systems. Traffic data, even though it is encrypted, can be attacked by purposely designed attack methods. [30] presented the application of attack methods against the encrypted traffic data. Xu et al. analyzed vulnerabilities in machine learning-based urban scene segmentation for autonomous vehicles, proposing an effective attack method and adversarial training to enhance model robustness with small adversarial perturbations, validated on the Cityscape dataset [31]. Xiong et al. investigated an attack model that is applicable to multiple data sources including images from camera and point cloud data from Lidar. They proposed two novel multi-source adversarial sample attack models for autonomous vehicles: the parallel attack model and the fusion attack model. These models simultaneously target image and LiDAR perception systems, exploiting data correlation for efficient adversarial sample generation. Through real-data experiments, they demonstrate the superiority and effectiveness of their approach compared to existing methods. Additionally, they simulate attack scenarios in Vehicular Ad hoc Networks to evaluate their methods' performance [32].

Other studies suggest that attacks can be performed against transportation systems. Zhang et al. studied physical adversarial attacks targeting object detectors in real-world scenarios. They proposed an approach to learn an

optimal camouflage on vehicles that can hide it from image-based object detectors. Results demonstrate that the learned camouflage can successfully conceal vehicles from image-based detectors across various test cases and environments, showcasing its generalizability [33].

For systems coordinating vehicles, attacks can be injected as discussed in [34], which motivates the authors to propose a federated learning-based approach to defend against attacks. Vehicles controlled by malicious agents can cost significant losses in transportation systems. Thai et al. studied the scenario where a malicious agent controls a fraction of vehicles in MaaS to perform DoS attacks. They developed a framework to analyze denial-of-service attacks on MaaS systems. They modeled attacks on vehicle control, formulated a control problem, and simulated scenarios to estimate passenger loss. They find that the defense strategy costs approximately U.S. \$15 per unit attack to protect the MaaS system [5]. Munoz et al. addressed the challenge of integrating various transport modes into a unified ticketing system for MaaS, ensuring resilience against attacks. They proposed two smartphone-based methods capable of identifying transport modes, including trams, buses, subways, walking, and bicycles, while detecting GNSS-based attacks. Real-world measurements in Munich and Paris validate the effectiveness of the proposed methods, showcasing their ability to provide trustworthy identification of transport modes even under attack. This work highlights the vulnerability of smartphone-based ticketing to GNSS-based attacks and offers practical solutions to mitigate this vulnerability [35].

Zhou et al. introduced a novel resilience analysis method for MaaS reliability, addressing the lack of analytical approaches for the complex interactions within integrated services. Leveraging enterprise architecture modeling, they propose a comprehensive framework to assess MaaS resilience. Through a controlled experiment, they validate the efficacy of their method, providing a quantitative comparison with a referenced approach, thereby advancing the understanding and development of reliable mobility services [36]. Chu et al. proposed a federated deep deterministic policy gradient (FDDPG) method to optimize MaaS systems while preserving passenger privacy. Unlike centralized methods, FDDPG ensures local information processing and gradient sharing to prevent data leakage. Experiments demonstrate significant improvements in MaaS profit and passenger satisfaction, enhancing trust in data-driven transportation [37].

2.5 Research Gap

As reviewed in the previous subsections, there are many AI security threats and potential attacks against transportation systems. The situation could be even worse for MaaS with advanced cyber technologies, which is one of the intelligent transportation systems that may be implemented in the near future. In the context of passenger heterogeneity in MaaS, the existing literature is currently lacking. While there have been studies on reinforcement learning-based attacks, such as the work discussed in [38], these primarily focus on data poisoning attacks in crowdsensing systems rather than addressing the specific threat to passenger heterogeneity in

MaaS. Chen et al. [39] and Ilahi et al. [40] have conducted reviews on adversarial attacks and defense in reinforcement learning, but none of the attacks examined align precisely with the attack described in this paper. Although Tu et al. [41] proposed a detection method for impersonation attacks, it is important to note that their scenario pertains to device-to-device communications rather than transportation systems.

Regarding spoofing attacks, existing research has primarily focused on spoofing attacks related to Global Positioning System [42] and Global Navigation Satellite System [43] in transportation systems, rather than targeting passenger information specifically. Therefore, we believe that the security of smart MaaS and a possible new attack method should be re-evaluated in this paper to avoid serious security loopholes that may cause countless economic losses.

Table 1 summarizes the vulnerabilities, potential consequences, and targets of transportation systems in the recent literature.

3 MAAS JOURNEY PLANNING

In this section, we introduce the background of the scenario including the MaaS model and journey planning problem. The notation representation is summarized in Table 2.

3.1 MaaS Model

MaaS integrates multiple mobility providers that provide mobility services for the same or different routes. A MaaS coordinator is responsible for coordinating all the mobility services of mobility providers so that passengers can easily select, reserve, and pay for the combined journey traversing their origin and destination. We consider the multimodal transport network as a directed graph $G(\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ and $\mathcal{A}$ is the set of nodes and links in the network, respectively. Let $\mathcal{F}$ be the set of mobility providers. Each mobility provider $f \in \mathcal{F}$ offers mobility services on a sub-network $\mathcal{A}_f \in \mathcal{A}$. Any mobility service from node i to node j offered by f is associated with utility costs, including time¹, discomfort, price, and operation cost that are represented by β_{ij}^f , δ_{ij}^f , ρ_{ij}^f , and μ_{ij}^f , respectively. The duty of the MaaS coordinator is to determine an optimal journey fulfilling passenger k transport request from origin o^k to destination d^k based on the passenger preferences, experiences, and memories. The MaaS coordinator considers the mobility services of multiple mobility providers for multiple passengers.

3.2 Journey Planning Problem

The optimal journey planning problem is presented in this section.

To facilitate the formulation, we define two binary variables, x_{ij}^{kf} and y_{ij}^f , to represent the coordinator decision of

1. In a real-world application, travel time may include waiting, in-vehicle, and transfer time. As a pilot study and for simplicity, we use a single parameter rather than a series of time-related parameters to represent all the travel time cost. That parameter can be an aggregation of the detailed time parameters

TABLE 1
The vulnerabilities, potential consequences, and targets of transportation systems.

References	Studied vulnerabilities	Potential consequences	Targets
[16]	Information transfer	Data leakage	Communication between mobility providers and municipal authorities
[31]	The AI model is sensitive to adversarial perturbations	Incorrect urban scene segmentation	Autonomous Vehicles
[32]	The AI model is sensitive to adversarial perturbations	Incorrect object detection and classification	Autonomous Vehicles
[5]	Denial-of-Service attacks	Service failure	MaaS
[37]	Information transfer	Data leakage	MaaS
[35]	GNSS-Based Attacks	Incorrect location information	MaaS
[36]	Complexity of the interactions of its subsystems	Service failure	MaaS
This paper	The AI model is sensitive to adversarial perturbations	Impersonating priority passengers with falsified information	MaaS

TABLE 2
Notation summary.

Notations	Meaning
$G(\mathcal{N}, \mathcal{A})$	Graph of the transport network
$\mathcal{N}$	Set of nodes in the network
$\mathcal{A}$	Set of links in the network
$\mathcal{F}$	Set of mobility providers
f	Mobility provider $f \in \mathcal{F}$
$\mathcal{N}^+(i)$	Set of incoming locations of i
$\mathcal{N}^-(i)$	Set of outgoing locations of i
$\mathcal{A}_f$	Subset of $\mathcal{A}$ operated by $f \in \mathcal{F}$
$\mathcal{K}$	Set of passengers
k	Passenger $k \in \mathcal{K}$
o^k	Origin node of passenger $k \in \mathcal{K}$
d^k	Destination node of passenger $k \in \mathcal{K}$
x_{ij}^{kf}	Binary variable for link $(i, j) \in \mathcal{A}_f$ that recommends to passenger $k \in \mathcal{K}$
y_{ij}^f	Binary variable for an operator to serve a link $(i, j) \in \mathcal{A}_f$
β_{ij}^f	Travel time cost of link $(i, j) \in \mathcal{A}_f$
w_β^k	Weight of β_{ij}^f
δ_{ij}^f	Discomfort index of link $(i, j) \in \mathcal{A}_f$
w_δ^k	Weight of δ_{ij}^f
ρ_{ij}^f	Price of link $(i, j) \in \mathcal{A}_f$
w_ρ^k	Weight of ρ_{ij}^f
μ_{ij}^f	Operating cost of link $(i, j) \in \mathcal{A}_f$
w_μ^k	Weight of μ_{ij}^f
C_{ij}^f	Capacity of link $(i, j) \in \mathcal{A}_f$

the journey. x_{ij}^{kf} represents the set of mobility services to be offered:

$$x_{ij}^{kf} = \begin{cases} 1 & \text{if link } (i, j) \text{ operated by } f \text{ offers to } k, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

y_{ij}^f represents the operation status of each mobility ser-

vice operated by mobility provider f .

$$y_{ij}^f = \begin{cases} 1 & \text{if link } (i, j) \text{ is operated by } f, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The objective function of the problem can be defined as minimizing the total utility cost of passengers and mobility providers, which is specified by

$$\sum_{(i,j) \in \mathcal{A}_f, k \in \mathcal{K}, f \in \mathcal{F}} (w_\beta^k \beta_{ij}^f + w_\delta^k \delta_{ij}^f + w_\rho^k \rho_{ij}^f) x_{ij}^{kf} + w_\mu^f \mu_{ij}^f y_{ij}^f, \quad (3)$$

where $w_\beta^k, w_\delta^k, w_\rho^k, w_\mu^f$ are the weighting of the corresponding utility terms.

Let $\mathcal{N}^+(i)$ and $\mathcal{N}^-(i)$ be the sets of incoming and outgoing locations of i , respectively, such that $\mathcal{N}^+(i) = \{j \in \mathcal{N} | (j, i) \in \mathcal{A}_f\}$ and $\mathcal{N}^-(i) = \{j \in \mathcal{N} | (i, j) \in \mathcal{A}_f\}$. The offered journey in the transport network has to be on a connected path which can be ensured by the flow conservation equation:

$$\sum_{j \in \mathcal{N}^-(i)} x_{ij}^{kf} - \sum_{j \in \mathcal{N}^+(i)} x_{ji}^{kf} = \begin{cases} 1 & \text{if } i = o^k, \\ -1 & \text{if } i = d^k, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}, k \in \mathcal{K}, f \in \mathcal{F}. \quad (4)$$

The total number of services to be offered is limited by the capacity. Thus, we restrict the total number of passengers for a service using the following equation:

$$\sum_{k \in \mathcal{K}} x_{ij}^{kf} \leq C_{ij}^f y_{ij}^f, \quad \forall (i, j) \in \mathcal{A}_f, f \in \mathcal{F}, \quad (5)$$

where C_{ij}^f is the maximum capacity of mobility service from i to j that operated by $f \in \mathcal{F}$.

As one may notice, the weights in Eq. 3 may affect the optimal solution obtained. Determining a proper set of weighting based on individual passenger preferences, experiences, and memories is of great interest to increase passenger satisfaction and MaaS profit for the journey planning problem. To capture the passenger experience and memories, a 4-tuple Markov decision process, $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$, can be used to model the passenger retention process, where

S , $\mathcal{A}$, P , and R are the set of states and actions, state transition and reward function, respectively. For each travel time t , the state $s_t^k \in \mathcal{S}$ represents passenger satisfaction and characteristics. The action $a_t^k \in \mathcal{A}$ is the utility weighting, where $a_t^k := [w_\beta^k; w_\delta^k; w_\rho^k; w_\mu^k]$ for time t and passenger k . $P(s_{t+1}^k | s_t^k, a_t^k)$ is the satisfaction transition from states $s_t^k \in \mathcal{S}$ to $s_{t+1}^k \in \mathcal{S}$ with action $a_t^k \in \mathcal{A}$. $R(s_t^k, a_t^k, s_{t+1}^k)$ can be used to model the profit for transiting from s_t^k to s_{t+1}^k by acting a_t^k :

$$R(s_t^k, a_t^k, s_{t+1}^k) = \sum_{f \in \mathcal{F}} (\rho_t^{kf} - \mu_t^{kf}). \quad (6)$$

We model the passenger satisfaction level by a N -level integer value, which is proportional to the retention rate. Let H^k be the satisfaction level of passenger k . Intuitively, satisfaction increases if the passenger is satisfied with the offered journey and decreases otherwise. Therefore, we model the satisfaction variation as a function based on the expectation difference between the expected and actual offered journey:

$$H^k := \begin{cases} H^k + n & \text{if } E^k \geq \bar{E}^k, \\ H^k - n & \text{if } E^k \leq \underline{E}^k, \\ H^k & \text{otherwise,} \end{cases} \quad \forall k \in \mathcal{K}, \quad (7)$$

where $\bar{E}^k$ and $\underline{E}^k$ is the upper and lower threshold of expectation difference, n is the step size of the satisfaction level, and the expectation difference E^k is defined as follows:

$$E^k = \tilde{w}_\beta^k (\tilde{\beta}_{o^k d^k}^k - \beta_{o^k d^k}^k) + \tilde{w}_\delta^k (\tilde{\delta}_{o^k d^k}^k - \delta_{o^k d^k}^k) + \tilde{w}_\rho^k (\tilde{\rho}_{o^k d^k}^k - \rho_{o^k d^k}^k), \quad \forall k \in \mathcal{K}, \quad (8)$$

where $\tilde{w}_\beta^k$, $\tilde{w}_\delta^k$, and $\tilde{w}_\rho^k$ are the corresponding utility weighting of passenger k , and $\tilde{\beta}_{o^k d^k}^k$, $\tilde{\delta}_{o^k d^k}^k$, $\tilde{\rho}_{o^k d^k}^k$ are the utility expected implicitly by the passenger. Note that the expected utility is the utility of the ideal best journey the passenger can get without capacity constraint, which can be determined by solving Eqs. (3) and (4) only. The actual utility is the utility of the journey planned by MaaS.

As a whole, the journey planning and passenger satisfaction problem is formulated as a bi-level problem:

Problem 1 (Journey Planning and Passenger Satisfaction Problem).

$$\begin{aligned} & \min_{x_{ij}^{kf}, y_{ij}^f} \quad (3) \\ & \text{s.t.} \quad (4)-(5), \\ & \quad a_t^k \in \arg \max_{k,t} \left\{ \sum_{k,t} R(s_t^k, a_t^k, s_{t+1}^k) : (7)-(8) \right\}. \end{aligned}$$

To solve this problem, one may use the method as proposed in [2]. The algorithm that solves this problem is out of the scope of this paper.

To summarize the MaaS model and the journey planning problem, one may consider that the MaaS operator solves Problem 1 to determine the optimal journey for passengers. Binary variables x_{ij}^{kf} in Eq. (1) and y_{ij}^f in Eq. (2) are the journey and operation status, respectively. To determine the value of these variables, Eq. (3) defines the objective function to be minimized. Eqs. (4) and (5) ensures the journey is not violating the flow and capacity constraints. In the lower-level optimization task, the MaaS operator maximizes the

profit defined in Eq. (6) with the satisfaction and expectation model in Eqs. (7) and (8), respectively. We would like to emphasize that maximizing the profit does not necessarily mean that the MaaS operator is taking advantage of the passengers. Indeed, because of the passenger retention model, the MaaS operator will offer the best journey to the passenger so that they will be retained in the system for higher future profit.

4 PASSENGER SPOOFING ATTACK

In this section, we present the threat model, problem formulation, and methodology of the PSA.

4.1 Assumptions

We assume that the attacker can acquire the service request of the target passenger and the system has limited transportation resources. We also assume a malicious agent tries to spoof the MaaS system so that the services to be provided to other regular passengers are expected to be devalued. To do this, there are two important components for the malicious agent. First, the malicious agent requests a service with the same or similar origin and destination as the target passenger. Second, a fake state (passenger satisfaction and characteristics) of the malicious agent is submitted to the system together with the service request. Therefore, the core of the attack is to design a mechanism to produce the fake state that is effective in decreasing the MaaS system performance such as profit and passenger satisfaction.

In an urban transportation system with heavy traffic and limited capacity, especially during rush hours, regular passengers with the same or similar travel path as the malicious agent could be offered a detour journey because of the attacks. Thus, passenger satisfaction and MaaS profit may be reduced due to the attacks. Moreover, a set of malicious agents may submit queries at the same time to enhance the attacking power. Hence, the scenario becomes a multi-agent setting where each agent may attack the MaaS system cooperatively. In this case, the agents are assumed to be cooperative in order to maximize the attacking power. We assume that the agent has no prior knowledge of the coordinator and treats it like a black box, i.e., the malicious agent does not know the formulation of Problem 1. Only the state of other regular passengers is known to the agent as the basic knowledge of the environment to generate a spoofing state, which can be obtained by eavesdropping or related attacks.

4.2 Problem Formulation of PSA

In a PSA, a set of malicious agents $\mathcal{M}$ intend to sabotage the MaaS operations or take advantage by spoofing to be another passenger. In an urban transportation system with limited capacity and resources, competing passengers may reduce the chances of getting the most desired services for other passengers. Based on this consequence, a malicious agent may act as passengers with similar origin-destination patterns, such as the daily home-to-work flows during rush hour, to occupy the resources of regular passengers. To maximize the attack influence, the malicious agent will generate characteristics and satisfaction based on the characteristics

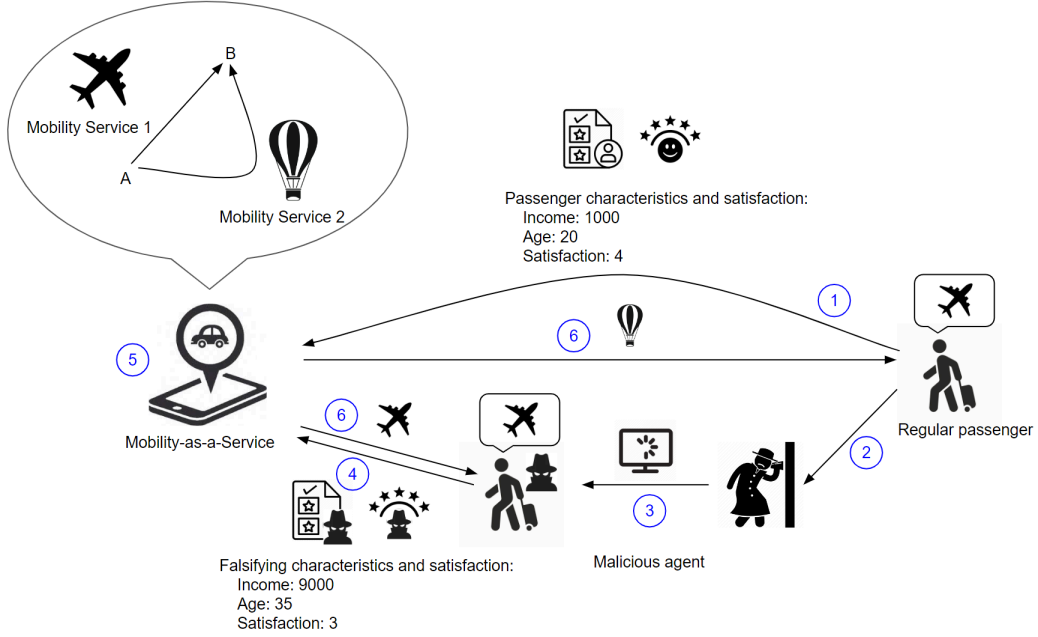


Fig. 1. An example showing the interactions among MaaS coordinator, regular passenger, and malicious agent.

and satisfaction of another passenger, which can be defined as:

$$\tilde{s}_t^m = \pi(s_t^k | \theta^\pi), \quad (9)$$

where s_t^k is state of passenger k including characteristics (income, age, etc.²) and satisfaction (H^k), $\tilde{s}_t^m$ is the generated state based on passenger k , π is the generation function, and θ^π is the trainable parameters of π .

The objective of the malicious agent is to minimize the profit as well as the satisfaction of the regular passengers. Therefore, we define the objective function of the PSA problem as the profit produced by the set of regular passengers $\mathcal{K}$ being attacked, which is represented as:

$$\sum_{k \in \mathcal{K}, t} R(s_t^k, a_t^k, s_{t+1}^k). \quad (10)$$

Since the profit is directly related to the performance of the MaaS coordinator as discussed in Section 3, the malicious agent should include the journey planning problem, as a black box, into the attack problem, which is formulated as:

Problem 2 (PSA Problem).

$$\begin{aligned} \min_{\theta^\pi} \quad & \sum_{k \in \mathcal{K}, t} R(s_t^k, a_t^k, s_{t+1}^k) \\ \text{s.t.} \quad & a_t^k \in \arg \max \left\{ \sum_{k \in \mathcal{K}, t} R(s_t^k, a_t^k, s_{t+1}^k) \right. \\ & \left. + \sum_{m \in \mathcal{M}, t} R(\tilde{s}_t^m, a_t^m, \tilde{s}_{t+1}^m) : (7), (8), (9) \right\}. \end{aligned}$$

The only controllable parameter in Problem 2 is θ^π . In other words, the problem is to minimize the profit of the set of passengers $\mathcal{K}$ as in Eq. (10) by determining the parameter θ^π and generation function π as shown in Eq. (9). This is different from the lower-level optimization task in Problem

1 that aims to maximize the profit of all the passenger $\mathcal{K}$ including the set of malicious agent $\mathcal{M}$. As a result, the malicious agent entices the MaaS coordinator to obtain more profit from $\mathcal{M}$. In comparison, the actual profit received from the regular passengers $\mathcal{K}$ will be lower. An example operation and flow are shown in Fig. 1. The malicious agent spoofed as a passenger with the same origin and destination as the regular passenger. The MaaS coordinator allocates the ideal mobility service to the malicious agent instead of the regular passenger based on falsifying characteristics and satisfaction. The detailed sequence of events is as follows:

1. A regular passenger sends a request to MaaS.
2. A malicious agent eavesdrops the information.
3. The malicious agent computes the falsifying information.
4. The malicious agent sends the falsifying request to the MaaS.
5. The MaaS computes the mobility service offer for the regular passenger and malicious agent.
6. The MaaS releases the offers to them.

Table 3 shows sample requests submitted by regular passengers and malicious agents, where malicious agents spoofed as regular passengers with the same origin and destination and different characteristics and satisfaction. The asterisk marked at the passenger indicates that those passengers are malicious agents targeting the one without the asterisk mark.

Please note that although travel time and discomfort terms are not in the reward function (10), they are still considered in the MaaS system. In fact, the MaaS coordinator uses an additional journey planner to minimize passenger travel time and discomfort in Eq. (3), together with the agent with the profit-only reward. Therefore, travel time and discomfort are included in the proposed system.

2. Here, income and age are just example characteristics. It can represent other characteristics without loss of generality.

TABLE 3
Sample data submitted by regular passengers and malicious agents.

Passenger	From	To	Income	Age	Satisfaction
1	246	186	2994	43	5
1*	246	186	957	31	3
2	163	230	1886	37	2
2*	163	230	1263	51	4
3	114	229	454	24	4
3*	114	229	798	33	3

Algorithm 1 Multi-agent RL algorithm for PSA

```

1: Initialize actor local  $\pi(s|\theta^\pi)$  and critic local networks
    $Q(s, a|\theta^Q)$  with parameters  $\theta^\pi$  and  $\theta^Q$ 
2: Initialize parameters of actor target  $\pi'(s|\theta^{\pi'})$  and critic
   target networks  $Q'(s, a|\theta^{Q'})$  with parameters  $\theta^{\pi'} \leftarrow \theta^\pi$ 
   and  $\theta^{Q'} \leftarrow \theta^Q$ 
3: for each episode do
4:   Initialize passengers' parameters
5:   for iteration  $t = 1$  to  $T$  do
6:     for malicious agent  $m = 1$  to  $|\mathcal{M}|$  do
7:        $j^m \leftarrow$  random number between 0 and 1
8:       if  $j^m < \epsilon$  then
9:          $\tilde{s}_t^m \leftarrow$  random action between 0 to 1
10:      else
11:         $\tilde{s}_t^m \leftarrow \pi(s_t^k|\theta^\pi)$ 
12:      end if
13:    end for
14:    for malicious agent  $m = 1$  to  $|\mathcal{M}|$  do
15:      Execute action  $\tilde{s}_t^m$  to obtain new state  $s_{t+1}^m$  and
        reward  $\tilde{r}_t^m$ 
16:       $\tilde{r}^m := \sum_{d=0}^{D_m} \left( \sum_{j \in \mathcal{V} | d(m,j)=d} \alpha^d r^j \right)$ 
17:      Store transition  $(s_t^k, \tilde{s}_t^m, \tilde{r}_t^m, s_{t+1}^m)$  to replay buffer
         $\mathcal{R}$ 
18:    end for
19:    if number of transitions  $|\mathcal{R}| \geq$  minibatch size  $B$ 
        then
20:      Sample a random minibatch of  $B$  transitions
         $(s_i^m, \tilde{s}_i^m, \tilde{r}_i^m, s_{i+1}^m)$  from  $\mathcal{R}$ 
21:      Update critic local  $\theta^Q$  based on Eq. (12)
22:      Update actor local  $\theta^\pi$  based on Eq. (13)
23:      Update critic target  $\theta^{Q'}$  based on Eq. (14)
24:      Update actor target  $\theta^{\pi'}$  based on Eq. (15)
25:    end if
26:  end for
27:   $\epsilon := \epsilon \bar{c}$ 
28: end for
29: return actor parameters  $\theta^\pi$ 

```

4.3 Methodology of PSA

To determine the generation function π and parameters θ^π that generate falsifying characteristics and satisfaction based on regular passenger characteristics and satisfaction, the malicious agent may use a reinforcement learning approach to learn from the interactions between the MaaS coordinator and passengers. Moreover, in an intensive scenario, multiple malicious agents may perform the attack at the same time and the agents may affect each other in the MaaS. Therefore,

a multi-agent reinforcement learning algorithm that considers the spatial distribution among the malicious agents and passengers could be used to enhance the cooperativeness among the agents.

The algorithm is a modified version of deep deterministic policy gradient (DDPG) [44] to suit the PSA problem. It is a model-free algorithm for continuous control problems as the model of the MaaS coordinator is unknown to the malicious agent and the states and actions are continuous values. Since the malicious agent is in the same environment as the MaaS coordinator, it shares the same Markov decision process except with two differences from the perspective of the malicious agent. The first difference is that the action of the malicious agent is the generated characteristics and satisfaction for the passenger spoofing, i.e., $\tilde{s}_t^m$. The second difference is that it maximizes the reward function, which is equal to minimize the negative profit, i.e., $-\sum_{k \in \mathcal{K}, f \in \mathcal{F}, t} (\rho_t^{kf} - \mu_t^{kf})$. The algorithm contains two components, actor and critic. The actor function $\pi(s|\theta^\pi)$ is for performing an action based on a given state, and a critic function $Q(s, a)$ learns the Q-value of the state and action pair for evaluating the actor. The exploration of the algorithm is modified from an Ornstein-Uhlenbeck (OU) noise-based method to an epsilon-greedy action selection method to ensure that the action remains in the designated range. The epsilon-greedy action selection method specifies that a uniformly random action is generated under the probability represented by ϵ , where the value of ϵ decay along the episode for more exploitation in later episodes.

For multi-agent scenarios, it is designed in a distributed manner rather than a master agent controlling all the slaves. This design can effectively overcome the scalability issue for a large number of agents. However, the distribution of agents means that each agent is responsible for their local action but not a global consideration. Indeed, the action of one agent may affect not only the target passenger, but also other passengers nearby. Intuitively, the smaller the distance between the malicious agent and the passenger, the larger the attacking effect on the passenger. Therefore, to account for this observation, we introduce a spatial discount factor α , ranging between 0 and 1. The resulting reward function of each malicious agent is defined as:

$$\tilde{r}^m = \sum_{d=0}^{D_m} \left(\sum_{j \in \mathcal{V} | d(m,j)=d} \alpha^d r^j \right), \quad (11)$$

where $d(m, j)$ is the distance between agent m and passenger j and D_m is the maximum distance from agent m . α is used to scale down the reward of other passengers proportional to their distance. $\alpha = 0$ and $\alpha = 1$ are special cases for greedy control and cooperative control, respectively. The agents in the former case consider the local reward of the targeted passenger only, and the agents in the latter case consider the common global reward, which is the sum of all passengers.

The four function approximators in the algorithm are distinguished as actor local, actor target, critic local, and critic target networks. The update rules for these function approximators are presented as follows: The critic local

network θ^Q is updated based on the loss function:

$$L = \frac{1}{B} \sum_i (r_i + \gamma Q'(s_{i+1}, \pi'(s_{i+1} | \theta^{\pi'}) | \theta^{Q'}) - Q(s_i, a_i | \theta^Q))^2, \quad (12)$$

where B denotes the minibatch size, i represents the index of the sample in the minibatch, and γ signifies the discount factor. The actor local network θ^π is updated using policy gradient:

$$\nabla_{\theta^\pi} J \approx \frac{1}{B} \sum_i \nabla_a Q(s, a | \theta^Q) |_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s | \theta^\pi) |_{s_i}. \quad (13)$$

The utilization of local and target networks serves to stabilize the training process. In each iteration, the parameters of the local networks are updated according to the aforementioned rules. Conversely, the target networks are updated by softly copying the parameters from the local networks, with a scaling factor of τ . For the critic target $\theta^{Q'}$

$$\theta^{Q'} := \tau \theta^Q + (1 - \tau) \theta^{Q'}. \quad (14)$$

Similarly, for the actor target $\theta^{\pi'}$:

$$\theta^{\pi'} := \tau \theta^\pi + (1 - \tau) \theta^{\pi'}. \quad (15)$$

The detailed algorithm is given in Algorithm 1.

The most computationally intensive component of the model is the parameter updates of actor and critic local deep neural networks [45]. To focus solely on this component and simplify the analysis, other operations are ignored. In this simplified analysis, we assume that the actor and critic networks have the same number of layers and neurons, and we neglect the input dimension, which is typically smaller compared to the number of layers and neurons. Additionally, the activation function is not taken into account. The parameter updates involve calculating the loss function in Eq. (12) and the policy gradient in Eq. (13), which consist of the feedforward pass and backpropagation. The computational complexity of the feedforward pass is closely associated with the number of layers and neurons involved due to the matrix multiplication operations. For each training iteration and neural network layer, the complexity of matrix multiplication is $O(Bh^2)$ [46], where B is the batch size and h is the number of neurons. Considering a neural network with h inputs and outputs, the complexity becomes $O(Blh^2)$, where l is the number of layers. Since the input and output sizes are typically small in practice, we can simplify the complexity to $O(B((l-2)h^2 + 2h))$. The complexity of backpropagation is similar to that of the feedforward pass, as it involves similar computational processes but in reverse order. It can be expressed as $O(B((l-2)h^2 + 2h))$ [47] with the same simplification. Therefore, the total complexity for actor and critic updates per training iteration is $O(3B((l-2)h^2 + 2h))$, accounting for the feedforward pass, backpropagation in critic loss, and gradient update in the actor. Considering the number of iterations, T , and the number of malicious agents, M , the final complexity is $O(T * M * 3B((l-2)h^2 + 2h))$.

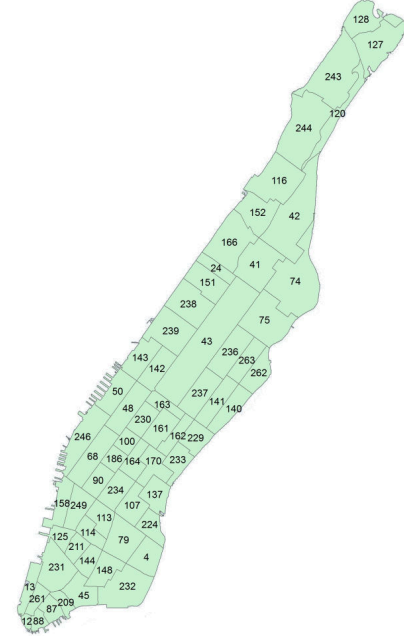


Fig. 2. NYC transport network.

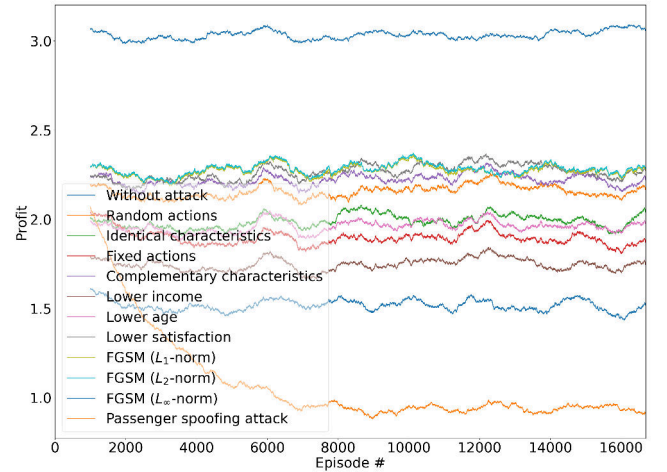


Fig. 3. Moving average reward of NYC scenario. The time window of the moving average is equal to 1000.

5 EXPERIMENTS

The reinforcement learning-based attack method is evaluated in two scenarios: a real-world transport network based on New York City (NYC) and a synthetic 6×6 grid. Both scenarios simulate the MaaS environment with multiple mobility providers and passengers. This section shows the experiment setups and results to validate the attack ability of MaaS.

5.1 Experiment Setups

5.1.1 New York City Scenario

A real-world scenario based on the transport network in NYC is constructed. The transport network is based on the

TABLE 4
Parameter settings.

Parameter	Definition	Value
NYC transport network and datasets		
$ \mathcal{N} $	Number of nodes	63
$ \mathcal{A} $	Number of links	963
$ \mathcal{F} $	Number of mobility providers	3
Synthetic transport network and datasets		
$ \mathcal{N} $	Number of nodes	36
$ \mathcal{A} $	Number of links	360
$ \mathcal{F} $	Number of mobility providers	6
$ \mathcal{R} $	Replay buffer size	10^6
B	Minibatch size	128
γ	Discount factor	0.99
τ	Target network soft update rate	0.001
-	Actor learning rate	0.0001
-	Critic learning rate	0.0003
-	Neural network optimizer	Adam
ϵ_0	Initial random explore rate	1
$\bar{\epsilon}$	Explore rate decay per episode	0.9995
T	Number of iteration per episode	10
l	Number of neural network layers	3
h	Number of neurons of each layer	256
-	Range of satisfaction level	1 to 5
$\bar{E}^k$	upper expectation threshold	0.0
$\underline{E}^k$	lower expectation threshold	-0.1

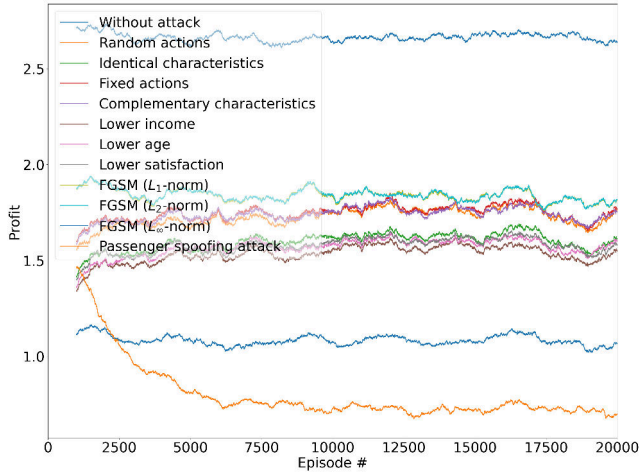


Fig. 4. Moving average reward of synthetic scenario. The time window of the moving average is equal to 1000.

taxi zone maps³ in the Manhattan region of NYC. A node denotes the taxi zone in the map, and an edge is assigned between two nodes if they are connected in the map. Only connected zone is included, and any isolated zones are ignored. The transport network contains 63 nodes in an irregular shape. 3 mobility providers are set for each edge. There are 963 edges in the transport network. Each mobility service has four attributes: time, discomfort, price, and

TABLE 5
Average profit of the baselines and attack of NYC and synthetic scenarios.

	NYC	Synthetic
Without attack	3.0431	2.6667
Random actions	2.1746	1.7505
Identical characteristics	2.0013	1.6205
Fixed actions	1.8927	1.7650
Complementary characteristics	2.2351	1.7567
Lower income	1.7592	1.5561
Lower age	1.9691	1.5907
Lower satisfaction	2.2891	1.5979
FGSM (L_1 -norm)	2.2722	1.8292
FGSM (L_2 -norm)	2.2796	1.8297
FGSM (L_∞ -norm)	1.5127	1.0804
PSA	0.9408	0.7227

operation cost. The time, discomfort, and price are randomly generated between 0 and 1. Operation cost is uniformly randomly generated between 0 and the corresponding price. Fig. 2 shows the taxi zone structure of the NYC scenario. The numbers on the taxi zones are the indexes defined in the dataset.

Passengers' traffic query and characteristics are simulated based on two open datasets of NYC: the NYC Taxi and Limousine Commission Trip Record Data⁴ and Citywide Mobility Survey⁵. The dataset is processed and filtered so that only people in the same Manhattan region as in the transport network are considered. The expected utility weight is determined based on the characteristics for the calculation of Eq. (8) only, which is unknown to the MaaS coordinator. All initial passenger satisfaction levels are at level 3. The multi-objective journey planning problem is solved by a standard optimizer in CVXPY [48].

5.1.2 Synthetic Scenario

A square grid with 6×6 nodes is used to simulate the transport network. Each node represents a stop or transit to other transport services. For two connected nodes, 6 edges are set where each edge denotes a mobility service operated by a different mobility provider. Hence, there are 360 edges in total among the nodes. Similar to the NYC scenario, the same attributes are assigned to each edge and generated using the same generator.

A set of passengers $\mathcal{K}$ with the random origin, destination, and characteristics is generated. All the values are normalized between 0 and 1. The expected utility weight, initial passenger satisfaction levels, and multi-objective journey planning problem-solving process are the same as the NYC scenario.

The parameter settings used in the experiments are shown in Table 4. We follow the network parameters of the dataset. For the model training parameters, it is based on common practice and general rule of thumb in the field of

4. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

5. <https://www1.nyc.gov/html/dot/html/about/citywide-mobility-survey.shtml>

3. <https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc>

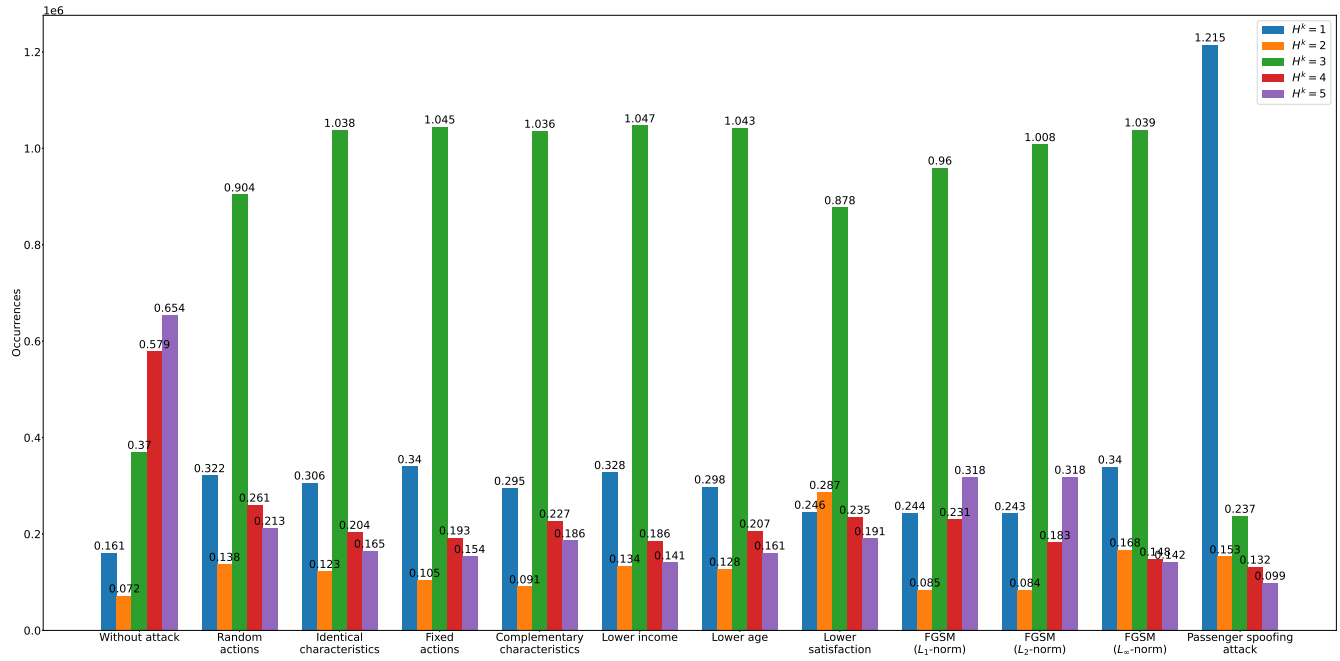


Fig. 5. Total number of satisfaction level of each method of NYC scenario.

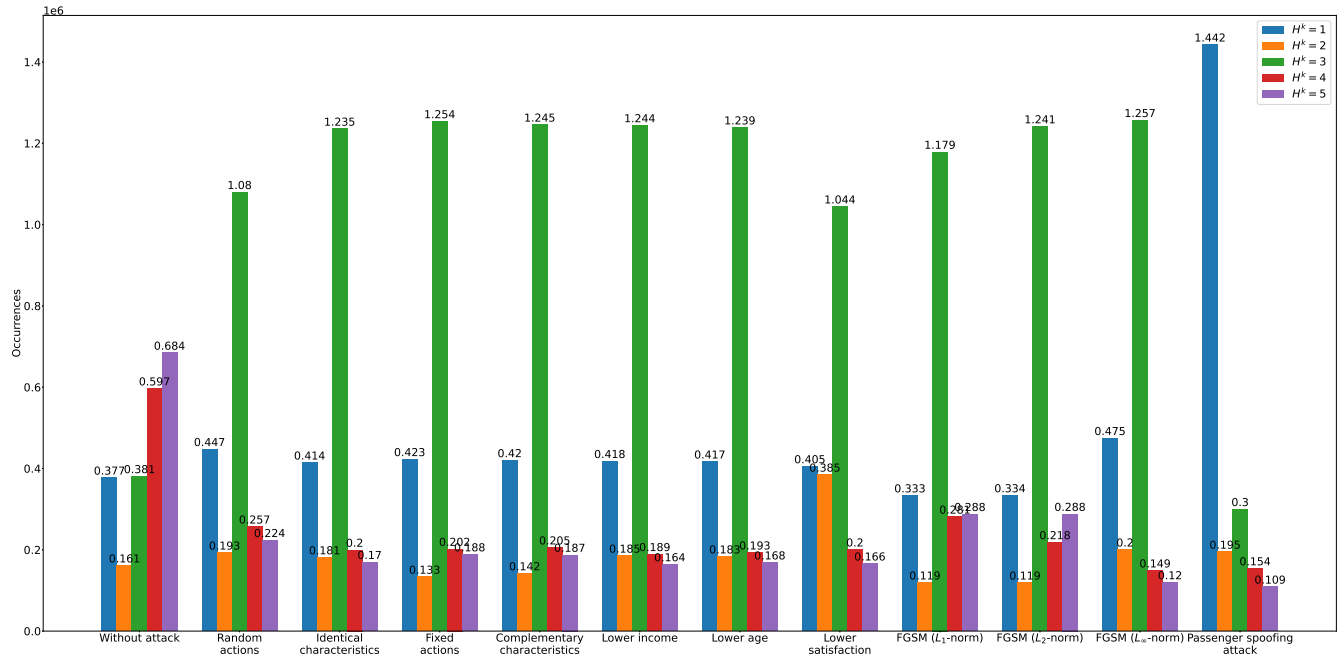


Fig. 6. Total number of satisfaction level of each method of synthetic scenario.

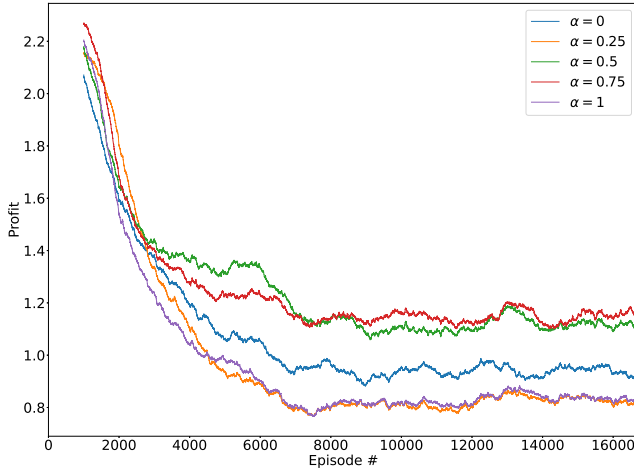


Fig. 7. Profit against episode with different spatial discount factor in the NYC scenario.

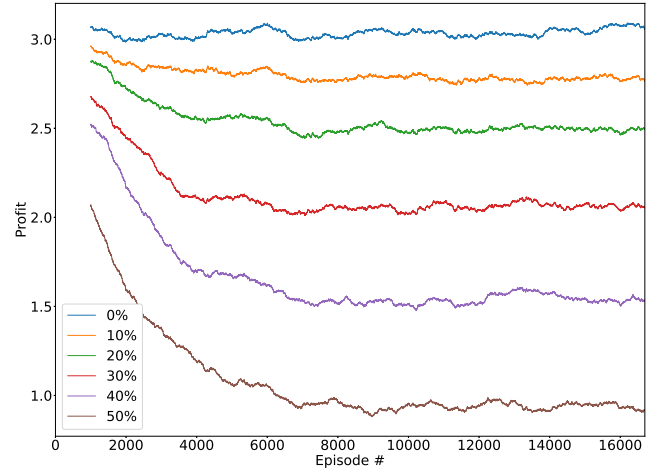


Fig. 9. Profit against episode with different proportion of malicious agents in the NYC scenario.

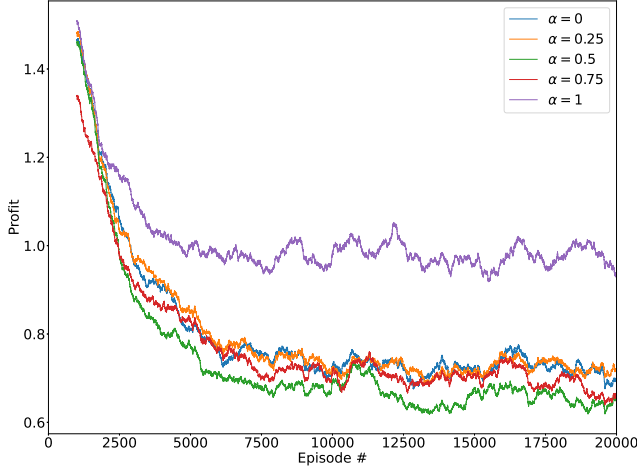


Fig. 8. Profit against episode with different spatial discount factor in the synthetic scenario.

machine learning with proper parameter tuning so that the training is optimized.

5.1.3 Baselines

We compare the reinforcement learning-based PSA with a few baselines as follows:

- Without attack: No malicious agent in this baseline. It is for comparing the performance level of the original smart system.
- Random actions: Malicious agents with randomly generated characteristics and satisfaction $\tilde{s}_t^m$ between 0 and 1.
- Identical characteristics: Malicious agents with the same characteristics and satisfaction $\tilde{s}_t^m$ as the target passenger.
- Fixed actions: Malicious agents with 0.5 characteristics and satisfaction $\tilde{s}_t^m$ in the normalized scale between 0 and 1.
- Complementary characteristics: The characteristics and satisfaction are computed based on one minus

target characteristics and satisfaction $\tilde{s}_t^m$ in the normalized scale between 0 and 1.

- Lower income: The income state of the malicious agents is always half of the target passenger's income in the normalized scale between 0 and 1.
- Lower age: The age state of the malicious agents is always half of the target passenger's age in the normalized scale between 0 and 1.
- Lower satisfaction: The satisfaction state of the malicious agents is always half of the target passenger's satisfaction in the normalized scale between 0 and 1.
- FGSM (L_1 -norm): Adversarial attacks based on [49]. A perturbation η is added to the input x of the policy leading to degraded performance. We use the policy gradient $\nabla_x J(\theta, x, y)$ for the computation of perturbation η instead of the gradient of the cross-entropy loss in [49], where η equals maximally perturb highest-impact dimensions with budget εd for constraint $\|\eta\|_1 \leq \|\varepsilon \mathbf{1}_d\|_1$, η is a small positive value set to be 0.05 in the experiments, and d is the number of dimensions of input x .
- FGSM (L_2 -norm): Similar to the above attack [49] except $\eta = \varepsilon \sqrt{d} \frac{\nabla_x J(\theta, x, y)}{\|\nabla_x J(\theta, x, y)\|_2}$ for constraint $\|\eta\|_2 \leq \|\varepsilon \mathbf{1}_d\|_2$.
- FGSM (L_∞ -norm): Similar to the above attack [49] except $\eta = \varepsilon \text{sign}(\nabla_x J(\theta, x, y))$ for constraint $\|\eta\|_\infty \leq \varepsilon$.

5.2 Experiment Results

We test and analyze the attack ability of the malicious agent in various aspects: the ability to reduce MaaS profit (1) and passenger satisfaction (2), the efficacy of the spatial discount factor in multi-agent setting (3), the sensitivity of the proportion of malicious agents (4) and the spatial distance from the regular passengers (5), and the distribution of actions (6).

5.2.1 Profit

We compare the baselines and PSA in each traffic scenario to analyze the MaaS profit. The baselines refer to the attacks

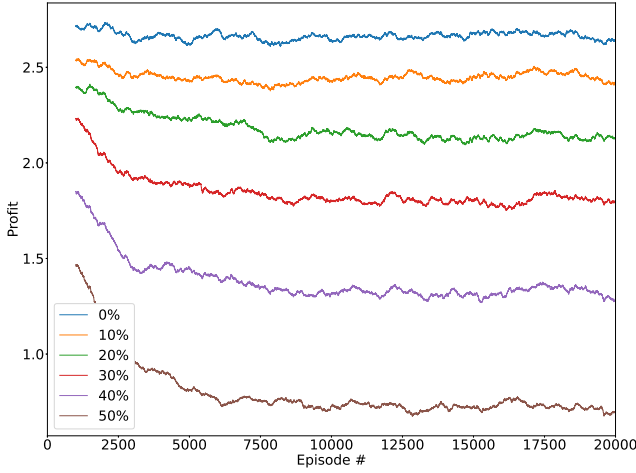


Fig. 10. Profit against episode with different proportion of malicious agents in the synthetic scenario.

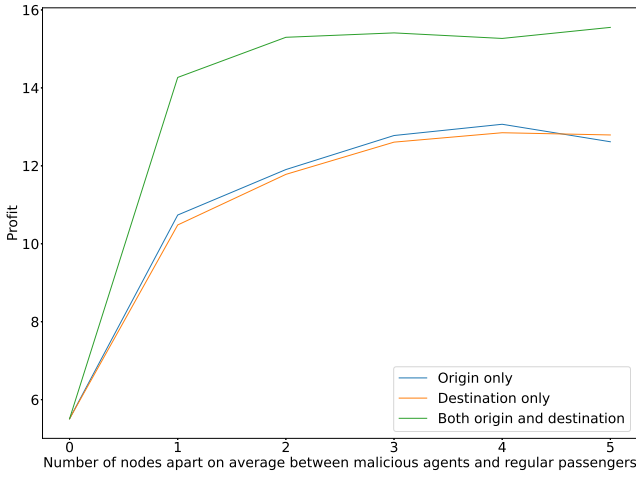


Fig. 11. Profit against number of nodes apart on average between malicious agents and regular passengers in the NYC scenario.

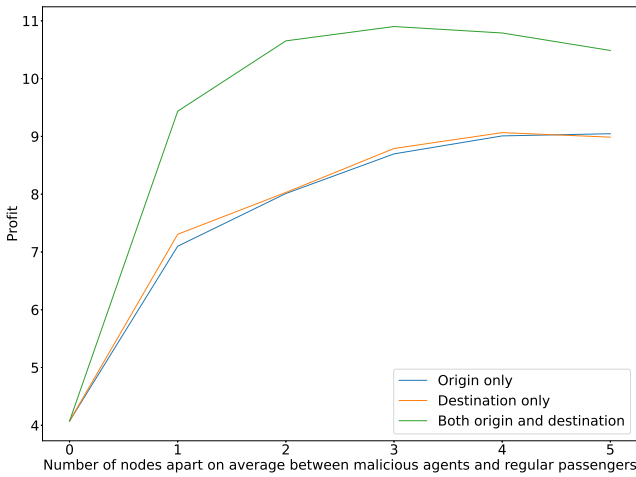


Fig. 12. Profit against number of nodes apart on average between malicious agents and regular passengers in the synthetic scenario.

as discussed in Section 5.1.3. The results of NYC and the synthetic scenario are shown in Figs. 3 and 4, respectively. The values of the lines are the profit of each episode received per regular passenger against the corresponding method during training, which means the lower the profit, the stronger the attack. To ensure consistency, there are 50% regular passengers and 50% malicious agents in all attack cases, except there are 100% regular passengers for the one without attack. The total number of regular and malicious passengers with and without attacks are the same so it is a fair comparison.

Both traffic scenarios show a similar trend for all cases. The case without attack has the highest profit, while the one with PSA has the lowest profit. Most of the baselines are in the middle. The disparity observed between the scenarios without any attack and the baselines illustrate the impact of introducing incapable agents with matching origins and destinations. On the other hand, the distinction between the baselines and the PSA highlights the attack capability of the proposed method. Among the three FGSM attacks, L_∞ -norm has the largest influence on the profit, while the other two do not show a significant influence on the profit. The proposal attack demonstrates a strong attack over all the FGSM attacks.

Through the training process, the malicious agent is equipped to generate suitable characteristics and satisfaction levels that have the potential to receive higher priority in journey planning compared to other regular passengers. The profit drops from the beginning to about 7500 episodes and converges afterward. The average profit of the baselines and attack of both scenarios are given in Table 5. Since those attacks aim to decrease the profit, the lower the profit, the stronger the attack. For the NYC scenario, the average profit with and without PSA after 10,000 episodes are 0.9408 and 3.0431 units, respectively. Most of the other baselines of NYC scenario are between 1.7 to 2.2 units. As for the synthetic scenario, the average profit with and without attack after 10,000 episodes are 0.7227 and 2.6667 units, respectively. Most of the other baselines of the synthetic scenario are between 1.5 to 1.8 units. Both scenarios have about 70% profit reduction with the attacks. This suggests that the malicious agent can supplant regular passengers if the capacity is limited using the generated characteristics and satisfaction.

5.2.2 Satisfaction

We analyze the satisfaction difference in both scenarios for the baselines and PSA. Figs. 5 and 6 summarize the occurrence of regular passenger satisfaction of all episodes of NYC and synthetic scenario, respectively. Since the bars indicate the occurrences of different satisfaction level, the higher the bar of higher satisfaction levels represent better services. In general, both scenarios have a similar trend of occurrence. The leftmost bars in the figures represent the results obtained from the MaaS coordinator without any attack, revealing a significant number of high satisfaction levels. However, upon introducing the baselines, depicted in the middle bars, the satisfaction of regular passengers noticeably declines, with the majority reaching level 3. In the case of PSA, illustrated by the rightmost bars, the majority of passengers experience the lowest satisfaction level. These findings suggest that the malicious agent possesses

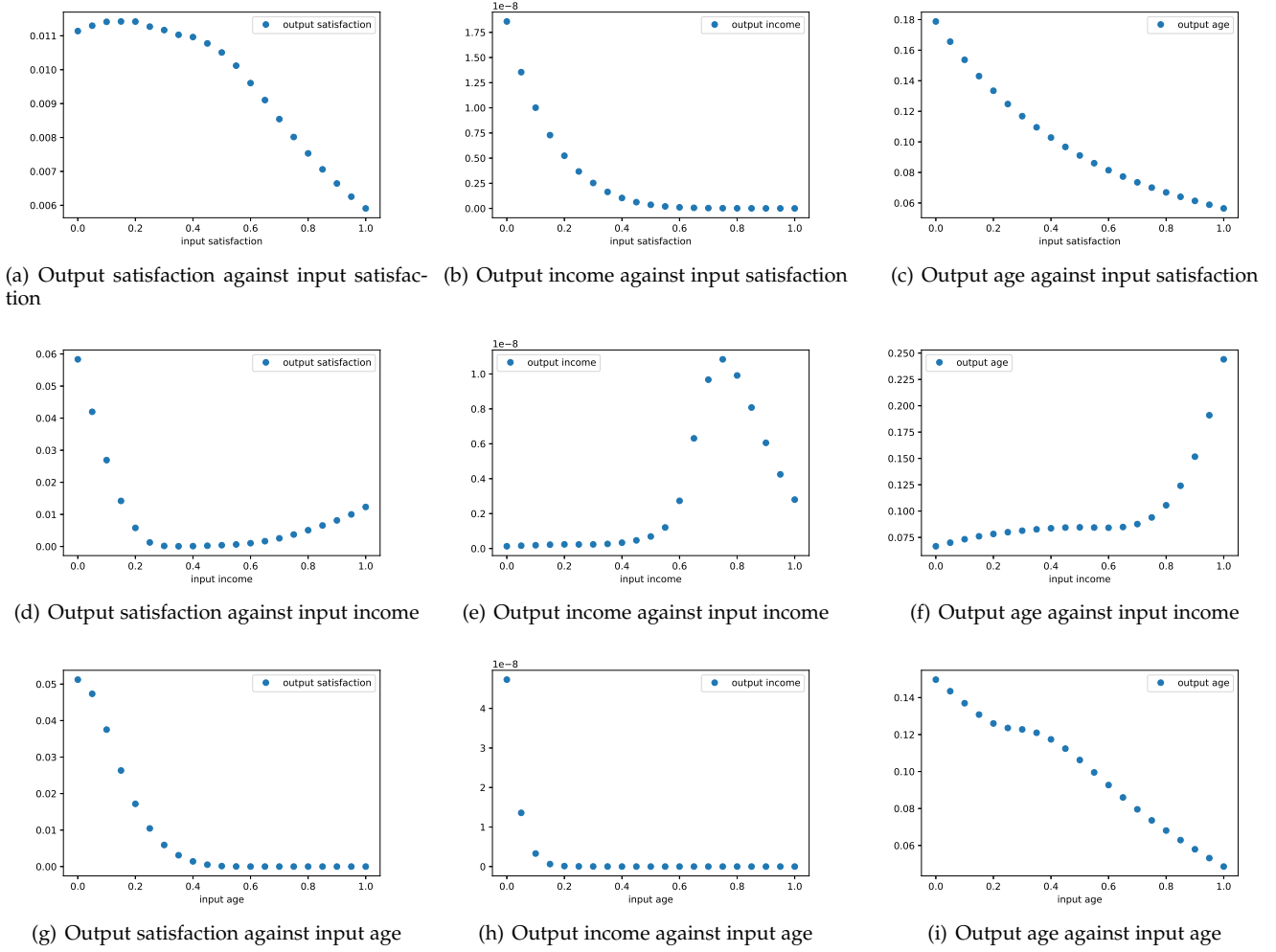


Fig. 13. Output actions against input states for NYC scenario.

the capability to diminish the satisfaction levels of other regular passengers by assuming the identity of a passenger with predetermined characteristics and satisfaction. This observation aligns with the outcomes observed in the profit analysis.

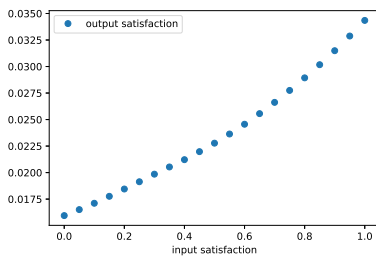
5.2.3 Multi-agent and spatial discount factor

We conduct the sensitivity analysis of the spatial discount factor α in multi-agent NYC and synthetic scenarios. Five cases of each scenario were performed with different values of α . The results of NYC and synthetic scenarios are given in Figs. 7 and 8, respectively. The values of the lines are the profit of each episode, which means the lower the profit, the stronger the attack. The two scenarios exhibit distinct patterns in their results. In the NYC scenario, the values of α that yield the best performance are 0.25 and 1, while a value of 0.5 yields the worst performance. Conversely, in the synthetic scenario, a value of 0.5 produces the best performance, while a value of 1 results in the worst performance. This contradictory outcome suggests that there is no universally optimal value of α that can guarantee the best performance across all scenarios. Similar to other parameters, the impact of α and its optimal value vary depending on the specific case. However, the introduction of

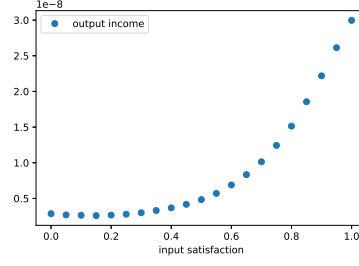
the spatial discount factor α affords flexibility in achieving higher performance. This feature represents a generalization of the single-agent algorithm and can potentially enhance the attack's effectiveness through appropriate parameter tuning.

5.2.4 Malicious agent proportion

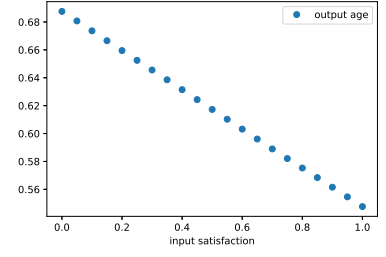
We are interested in how the proportion of malicious agents affects the MaaS, and thus we vary the proportion in each scenario. Figs. 9 and 10 plotted the profit along episodes with different proportions of malicious agents. The values of the lines are the profit of each episode, which means the lower the profit, the stronger the attack. In general, both scenarios exhibit a consistent pattern, as evidenced by the observed results. As anticipated, an increase in the number of malicious agents results in a decrease in profit. Interestingly, the decline in profit is not linearly proportional to the number of malicious agents. Instead, the reduction becomes more pronounced as the proportion of malicious agents increases. This suggests that the presence of malicious agents has a spreading and compounding effect on other passengers, further exacerbating the negative impact on overall profit.



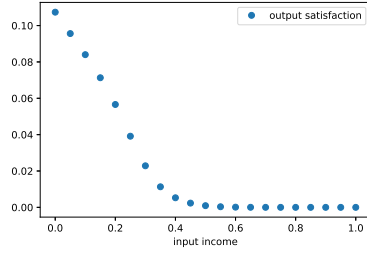
(a) Output satisfaction against input satisfaction



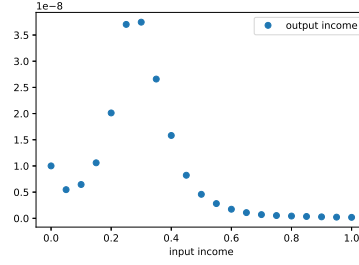
(b) Output income against input satisfaction



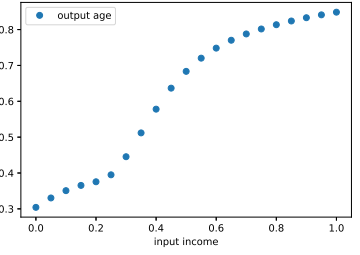
(c) Output age against input satisfaction



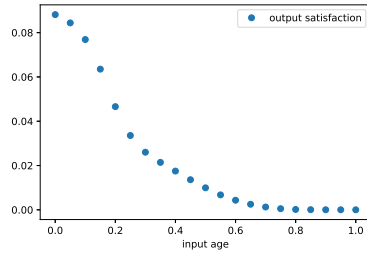
(d) Output satisfaction against input income



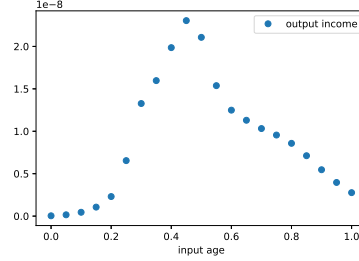
(e) Output income against input income



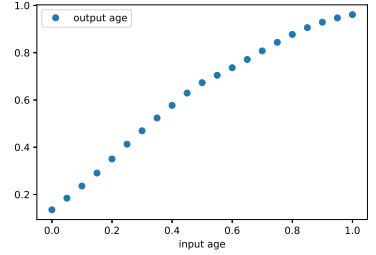
(f) Output age against input income



(g) Output satisfaction against input age



(h) Output income against input age



(i) Output age against input age

Fig. 14. Output actions against input states for synthetic scenario.

5.2.5 Spatial effect between malicious agent and regular passenger

As indicated from the analysis of malicious agent proportion, the attack has a spreading effect to regions other than the malicious agent's origin and destination. Hence, we are interested in how the spatial distance from the passengers relates to the attack ability. We vary the origin and/or destination of the malicious agent and plotted the profit against the number of nodes apart for origin only, destination only, and both origin and destination in Figs. 11 and 12 for NYC and synthetic scenario, respectively. The values of the lines are the profit of each episode, which means the lower the profit, the stronger the attack. Generally, there is an upward trend in profit as the number of nodes increases. Notably, the green line, representing variations in the number of nodes between the origin and destination, exhibits a greater increase compared to the other two cases involving either the origin or destination alone. In the case where both the origin and destination are varied, the profit increase begins to converge when the number of nodes apart exceeds one. On the other hand, for the other two cases, convergence occurs when the number of nodes apart exceeds three. It is worth noting that the impact of increasing the distance from a malicious agent is relatively minimal when the distance

reaches three nodes. Therefore, we may conclude that the effective distance is about two nodes from the origin or destination.

5.2.6 Attack distribution

We plot the distribution between output actions and input states of a well-trained malicious agent. We sample the output by inputting values from 0 to 1 in a step size of 0.05 for each element, including satisfaction, income, and age. Since the dimension is large for plotting, we compute the average output value of each element and show it in Fig. 13 and 14 for the agent of the NYC and synthetic scenario, respectively. In other words, we see what the expected output will be with different inputs. In general, the age feature exhibits a higher magnitude compared to the other features in both scenarios. However, caution must be exercised in concluding that age is the most influential factor in the prioritization of the AI-based model, as even small changes can significantly impact the model's outcomes, similar to the concept elucidated in the fast gradient sign method [19]. Furthermore, the relationships between the features and the prioritization outcome can be classified into positive, negative, and bell-shaped patterns. Nonetheless, no consistent pattern emerges from the distribution in both

scenarios. Therefore, we posit that further investigation into the detector is needed to identify attacks based solely on the data. Implementing external security measures such as prior ID checks could be a viable approach to mitigate such attacks.

6 CONCLUSION

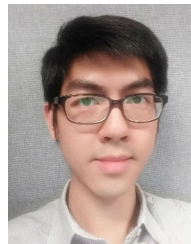
Operational failure of a transportation system may cause countless economic losses. A secure and trustworthy MaaS system is a backbone supporting the daily operation of an urban city. To prevent the MaaS from cybersecurity attacks, we must understand the vulnerability and identify possible attack methods. In this paper, we analyze the threat in the MaaS journey planning process. With the observation, we formulate the PSA problem that takes advantage of the heterogeneity of passengers. An algorithm based on multi-agent reinforcement learning is discovered for PSA to generate spoofing passenger profiles that are likely to be prioritized by the MaaS coordinator. To incorporate the interactions among the agents, a spatial discount factor can be introduced to the reward function to scale the reward of others based on the distance for strengthening the attack with a proper parameter setting. Experiment results on two scenarios based on a real-world and synthetic dataset and transport network are obtained, which indicates that the PSA can successfully reduce 70% of profit and most of the passenger satisfaction to the lowest level. The sensitivity of malicious agent proportion and spatial distance were analyzed. The relationships among those entities are useful for developing corresponding detection and defense mechanism. Although the main focus of this paper is on MaaS, this attack can also be applied to other smart systems that take user heterogeneity into consideration. In other words, this attack does not apply to systems that consider users to be the same. In addition, the state of the target passengers must have been known to the malicious agent, which is another limitation of this attack.

There are several directions for future research. First, our study focuses on the attack and its property. To build a secure MaaS, developing a corresponding measure against the attack should be conducted. One may approach the problem from the assumptions. Second, the current attack method mainly utilizes the passenger profile only, while the network structure may also be vulnerable to traffic bottlenecks and profit. Third, the attack algorithm trains a deterministic policy based on the state. The stochastic method that introduces randomness to the attack may hardly be detected by the filtering mechanism.

REFERENCES

- [1] P. Jittrapirom, V. Caiati, A.-M. Feneri, S. Ebrahimigharehbaghi, M. J. Alonso González, and J. Narayan, "Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges," *Urban Planning*, vol. 2, no. 2, pp. 13–25, 2017.
- [2] K.-F. Chu and W. Guo, "Deep reinforcement learning of passenger behavior in multimodal journey planning with proportional fairness," *Neural Computing and Applications*, pp. 20221–20240, 2023.
- [3] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Computers in Industry*, vol. 100, pp. 212–223, 2018.
- [4] F. Callegati, S. Giallorenzo, A. Melis, and M. Prandini, "Cloud-of-things meets mobility-as-a-service: An insider threat perspective," *Computers & Security*, vol. 74, pp. 277–295, 2018.
- [5] J. Thai, C. Yuan, and A. M. Bayen, "Resiliency of mobility-as-a-service systems to denial-of-service attacks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 370–382, 2016.
- [6] K.-F. Chu and W. Guo, "Federated reinforcement learning for consumers privacy protection in mobility-as-a-service," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 4840–4846.
- [7] K.-F. Chu, A. Y. Lam, and V. O. Li, "Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3219–3232, 2020.
- [8] K.-F. Chu, A. Lam, and V. Li, "Traffic signal control using end-to-end off-policy deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7184–7195, 2022.
- [9] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.
- [10] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," *Advances in neural information processing systems*, vol. 30, 2017.
- [11] K.-F. Chu and W. Guo, "Passenger spoofing attack for artificial intelligence-based mobility-as-a-service," in *Proceedings on the IEEE 26th International Conference on Intelligent Transportation Systems*, 2023, pp. 1–7.
- [12] M. Kamargianni and M. Matyas, "The business ecosystem of mobility-as-a-service," in *transportation research board*, vol. 96. Transportation Research Board, 2017.
- [13] O. Dib, L. Moalic, M.-A. Manier, and A. Caminada, "An advanced ga-vns combination for multicriteria route planning in public transit networks," *Expert Systems with Applications*, vol. 72, pp. 67–82, 2017.
- [14] Y. Wang, X. Lin, F. He, and M. Li, "Designing transit-oriented multi-modal transportation systems considering travelers' choices," *Transportation Research Part B: Methodological*, vol. 162, pp. 292–327, 2022.
- [15] S. K. Khan, N. Shiwakoti, P. Stasinopoulos, and Y. Chen, "Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions," *Accident Analysis & Prevention*, vol. 148, p. 105837, 2020.
- [16] M. Tsao, K. Yang, S. Zoepf, and M. Pavone, "Trust but verify: Cryptographic data privacy for mobility management," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 50–61, 2022.
- [17] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.
- [18] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.
- [19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.
- [20] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.
- [21] —, "Towards evaluating the robustness of neural networks," in *2017 IEEE symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.
- [22] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2020.
- [23] M. Xue, C. He, J. Wang, and W. Liu, "One-to-n & n-to-one: Two advanced backdoor attacks against deep learning models," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1562–1578, 2022.
- [24] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen, and T. Chen, "Backdoor attacks against transfer learning with pre-trained deep learning models," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1526–1539, 2022.

- [25] J. Breier, X. Hou, M. Ochoa, and J. Solano, "Foobar: Fault fooling backdoor attack on neural network training," *IEEE Transactions on Dependable and Secure Computing*, 2022.
- [26] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.
- [27] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.
- [28] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 619–633.
- [29] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020.
- [30] Y. Ding, G. Zhu, D. Chen, X. Qin, M. Cao, and Z. Qin, "Adversarial sample attack and defense method for encrypted traffic data," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [31] X. Xu, J. Zhang, Y. Li, Y. Wang, Y. Yang, and H. T. Shen, "Adversarial attack against urban scene segmentation for autonomous vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4117–4126, 2020.
- [32] Z. Xiong, H. Xu, W. Li, and Z. Cai, "Multi-source adversarial sample attack on autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2822–2835, 2021.
- [33] Y. Zhang, H. Foroosh, P. David, and B. Gong, "Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild," in *International Conference on Learning Representations*, 2018.
- [34] W. Yamany, N. Moustafa, and B. Turnbull, "Oqfl: An optimized quantum-based federated learning framework for defending against adversarial attacks in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [35] E. Munoz Diaz, J. M. Rubio Hernan, F. Jurado Romero, A. Karite, A. Vervisch-Picois, and N. Samama, "Advanced smartphone-based identification of transport modes: Resilience under gnss-based attacks," *Future Transportation*, vol. 3, no. 2, pp. 568–583, 2023.
- [36] Z. Zhou, Y. Matsubara, and H. Takada, "Resilience analysis and design for mobility-as-a-service based on enterprise architecture modeling," *Reliability Engineering & System Safety*, vol. 229, p. 108812, 2023.
- [37] K.-F. Chu and W. Guo, "Privacy-preserving federated deep reinforcement learning for mobility-as-a-service," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [38] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2019.
- [39] T. Chen, J. Liu, Y. Xiang, W. Niu, E. Tong, and Z. Han, "Adversarial attack and defense in reinforcement learning-from ai security view," *Cybersecurity*, vol. 2, pp. 1–22, 2019.
- [40] I. Ilahi, M. Usama, J. Qadir, M. U. Janjua, A. Al-Fuqaha, D. T. Hoang, and D. Niyato, "Challenges and countermeasures for adversarial attacks on deep reinforcement learning," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 90–109, 2021.
- [41] S. Tu, M. Waqas, S. U. Rehman, T. Mir, G. Abbas, Z. H. Abbas, Z. Halim, and I. Ahmad, "Reinforcement learning assisted impersonation attack detection in device-to-device communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1474–1479, 2021.
- [42] C. Vitale, N. Piperigkos, C. Laoudias, G. Ellinas, J. Casademont, J. Escrig, A. Kloukiniotis, A. S. Lalos, K. Moustakas, R. Diaz Rodriguez *et al.*, "Caramel: results on a secure architecture for connected and autonomous vehicles detecting gps spoofing attacks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–28, 2021.
- [43] S. Dasgupta, T. Ghosh, and M. Rahman, "A reinforcement learning approach for global navigation satellite system spoofing attack detection in autonomous vehicles," *Transportation research record*, vol. 2676, no. 12, pp. 318–330, 2022.
- [44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proceedings of the International Conference on Learning Representations*, 2016.
- [45] C. Li, A. Tsourdos, and W. Guo, "A transistor operations model for deep learning energy consumption scaling law," *IEEE Transactions on Artificial Intelligence*, 2022.
- [46] A. J. Stothers, "On the complexity of matrix multiplication," *Ph.D. Thesis, University of Edinburgh*, 2010.
- [47] J. J. Steil, "Backpropagation-decorrelation: online recurrent learning with $O(n)$ complexity," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2. IEEE, 2004, pp. 843–848.
- [48] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [49] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," in *Proceedings on the International Conference on Learning Representations*, 2017, pp. 1–10.



Kai-Fung Chu (S'17–M'20) received the Ph.D. degree in Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, in 2020, and the M.Sc. and B.Eng. (First Class Honors) degrees both in Electronic and Information Engineering from The Hong Kong Polytechnic University, Hong Kong, in 2016 and 2013, respectively. He is a Marie Skłodowska-Curie Fellow at the University of Cambridge. He was a Research Assistant Professor at the Hong Kong Polytechnic University and a Research Fellow at Cranfield University. He also worked in the industry as an engineer for several years. His research interests include machine intelligence, autonomous vehicles, and intelligent transportation systems.



Weisi Guo (M'11–SM'16) received his PhD degree from the University of Cambridge, UK in 2011. He was an Associate Professor at the University of Warwick (2012–19) and Turing Fellow (2017–19). He is currently a full Professor and head of the Human Machine Intelligence Group at Cranfield University, UK. He was a winner of the IET Innovation Award. His research interests focus on networks, data science, and autonomy.



DSPACE

<https://dspace.org/>

Multi-agent reinforcement learning-based passenger spoofing attack on Mobility-as-a-Service

Chu, Kai-Fung; Guo, Weisi

2024-11-01

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/TDSC.2024.3379283>

Chu KF, Guo W. (2024) Multi-agent reinforcement learning-based passenger spoofing attack on Mobility-as-a-Service. IEEE Transactions on Dependable and Secure Computing. Volume 21, Issue 6, November-December 2024, pp. 5565-5581