Distributionally Robust Cost Minimized Edge Semantic Intelligence in the Sustainable Metaverse

Wei Chong Ng, Wei Yang Bryan Lim, Zehui Xiong, Dusit Niyato, *Fellow, IEEE*,
Xuemin Sherman Shen, *Fellow, IEEE*, and Chunyan Miao, *Senior Member, IEEE*

***Abstract*—With the recent development of the Metaverse, people are more connected with each other. Avatars are used to represent the people, to communicate with one another, and they can build the community virtually. In these processes, a massive amount of data is exchanged between the physical and the virtual world. However, the existing communication technologies are insufficient to support the Metaverse, and the energy consumption of the Metaverse is huge. Therefore, semantic communication is one of the emerging communication paradigms to reduce the size of the data transmitted and reduce energy consumption while maintaining its meaning. Virtual service providers (VSPs) who provide services in the Metaverse can purchase semantic data from the nearby edge sensing units by using two subscription plans: reservation and on-demand. However, in practice, the demand of the VSPs is uncertain due to the variability of the Metaverse. To minimize the cost of the network and prevent over- and under-subscription of the resources, we propose a two-phase stochastic semantic resource allocation (SSRA) scheme. In phase one, a double dutch auction performs a one-to-one matching between VSPs and edge sensing units. The matching is dynamic and depends on the quality of experience (QoE) from the Metaverse users and the semantic data transmission cost from the edge sensing units. The matching changes whenever QoE and the semantic data transmission cost vary. In phase two, we consider the demand uncertainty and matching result from the phase one to formulate a distributed robust optimization (DRO) problem to minimize the operation cost of the VSPs. Using a real-world dataset, simulation results demonstrate that our proposed scheme is fully dynamic and minimizes the operation cost/energy consumption of VSPs in the presence of stochastic uncertainties.**

***Index Terms*—Metaverse, semantic communication, edge intelligence, incentive mechanism, resource allocation.**

I. Introduction

The Metaverse is the next generation of the Internet that allows users to experience the Internet of the future and has the potential to transform the way we live, interact, and learn [1]. It connects people around the world in the form of avatars to allow them to interact virtually through the use of virtual reality (VR)/augmented reality (AR)/mixed reality (MR) gadgets. Empowered by digital twins, the Metaverse closely synchronizes the virtual and physical world. For example, users can operate a virtual mechanical arm in the Metaverse, which is synchronized with the physical one located at a remote factory. With the help of VR/AR/MR, a user guide can be referred to by the user to operate the arm safely.

The Metaverse is expected to be built around seven key technologies: 6 G communication systems, extended reality, brain-computer interfaces, cloud/edge computing, blockchain, digital twins, and artificial intelligence (AI) [2]. Out of those emerging technologies, AI is the most important one in the Metaverse puzzle because of its ability to enable the Metaverse to scale. The AI-enabled Metaverse applications require a tremendous amount of data to train machine learning models so that the Metaverse applications can understand user inputs ranging from text to images and even videos and respond appropriately regardless of the user's input languages. In the future, machine learning models in the Metaverse will eventually be used to generate 3D images, animations, speech, and support blockchain technologies that allow for virtual transactions. However, the current Metaverse face with certain limitations, i) the Metaverse facilities require incredible amounts of energy, i.e., data centers rely on AI and cloud services to store information [3]. ii) The current 5 G communication system is insufficient to support the Metaverse, where millions or billions of devices are connected in the intersection between the virtual and physical world [4]. Given that current technologies are approaching the Shannon limit, traditional transmission methods are increasingly inadequate for meeting the rapidly growing and diversifying data traffic demands [5], [6].

In response, semantic communication enabled 6 G future communication systems may contribute to the development of

Manuscript received 27 December 2022; revised 26 September 2023; accepted 1 December 2023. Date of publication 18 December 2023; date of current version 5 June 2024. This work was supported in part by the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research & Development Programme, DSO National Laboratories under the AI Singapore Programme (AISG) under Grant AISG2-RP-2020-019, Energy Research Test-Bed and Industry Partnership Funding Initiative, Energy Grid (EG) 2.0 programme, DesCartes and the Campus for Research Excellence and Technological Enterprise (CREATE) programme, and MOE Tier 1 (RG87/22). Recommended for acceptance by A. Shami. *(Corresponding author: Wei Chong Ng.)*

Wei Chong Ng and Wei Yang Bryan Lim are with Alibaba Group and Alibaba-NTU Joint Research Institute, Nanyang Technological University, Singapore 637335 (e-mail: weichong001@e.ntu.edu.sg; limw0201@e.ntu.edu.sg).

Zehui Xiong is with the Singapore University of Technology and Design, Singapore 487372 (e-mail: zehui_xiong@sutd.edu.sg).

Dusit Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: dniyato@ntu.edu.sg).

Xuemin Sherman Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Chunyan Miao is with the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY) and School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ascymiao@ntu.edu.sg).

Digital Object Identifier 10.1109/TMC.2023.3343715

the sustainable Metaverse. In contrast to existing communication technologies, transmission in semantic communication is deemed effective if the received information maintains the same meaning as the transmitted information [7]. For instance, when a user requests an image, semantic communication systems reduce the transmitted data and transmit only the region in which the user is interested. As a result, the Metaverse energy consumption is reduced by reducing the transmitted data size.

In this paper, we propose a case study on the development of a virtual transportation network in the Metaverse that utilizes self-collected real-world data from selected streets in Singapore [8]. We consider a virtual service provider (VSP) that is an entity that offers a virtual service within the Metaverse. Using data from the physical domain, such as weather conditions or images of geographical landmarks and vehicles, the VSP can develop the Metaverse with the objective to provide users with immersive experiences, such as for realistic test driving of vehicles or the safe training of autonomous vehicles subject to practical constraints. Typically, data captured from physical domains can be traded on data markets or retrieved via crowdsensing [9]. Specifically, devices at the network's periphery may sell semantic data after it is collected and processed from the geographic regions in which they are deployed. The VSPs can purchase the semantic data transmissions from the edge sensing units by using two semantic data subscription plans, i.e., the reservation and on-demand plans [10]. The reservation plan is the long-term plan to purchase the semantic data that has been previously collected, processed, and stored by edge sensing units. On the other hand, the on-demand plan is an ad-hoc plan triggered whenever the VSPs require additional semantic data from the edge sensing units, e.g., when the VSP receives user feedbacks for low quality of experience (QoE) due to outdated physical-virtual world synchronization. As a result, the on-demand plan depends on parameters such as the uncertainty of QoE feedback from users, the interest of VSPs in the type of data required, and the number of data points required. To further reduce the energy consumption of the Metaverse, the cost function of the edge sensing units is formulated based on their energy consumption. Therefore, when the cost of the network is minimized, the energy consumption of the network is also minimized.

To derive the cost-minimizing semantic data subscription strategies of the VSP, we introduce a two-phase stochastic semantic resource allocation (SSRA). A two-phase SSRA is needed to allocate the semantic resource before and after observing the demand. In phase one, we first derive the expected on-demand cost for each corresponding set of uncertainty parameters. With our derived on-demand cost structure over the various uncertain scenarios, we formulate and solve for the subscription strategies of the VSPs as a distributed robust optimization (DRO) problem. The two phases are as follows.

- *Phase One (On-demand cost derivation):* Utilizing a single parameter set for each uncertainty scenario, we introduce a DRL-driven double dutch auction for one-to-one matching and price determination between VSPs and edge sensing units, thereby deriving the on-demand cost.
- *Phase Two (DRO for Subscription Strategy):* Addressing demand uncertainties in the Metaverse's cost minimization problem, we initially employ Stochastic Integer Programming (SIP) with historically-based random variables. Recognizing the limitations of SIP due to unpredictable distributions, we pivot to a more conservative, data-driven Distributed Robust Optimization (DRO) strategy. DRO constructs a modifiable confidence set from historical data to ensure network robustness, even when the true underlying distribution is unclear.

The contributions of this paper are summarized as follows.

- For the phase one of the SSRA scheme, our proposed DRL-driven-double dutch auction mechanism is tractable, dynamic, and adaptive to the stochastic network conditions. This enables us to devise the cost/energy consumption minimization strategy at scale. In addition, together with semantic communication enabled, a sustainable Metaverse can be realized.
- In the phase two of the SSRA scheme, instead of assuming the historical observation data is the true distribution, we construct a confidence set using the reference distribution derived from historical observation data and distance metrics. Our proposed DRO cost minimization formulation is able to mitigate the negative impacts of inaccurate uncertainty modeling of user demands in complex networks.
- Using our self-collected real-world data, we compare SSRA with other baselines, such as deterministic, random, and stochastic integer programming. From the performance evaluation, we demonstrate that SSRA can reduce significant amount of energy and achieve better outcomes relative to other baselines for extreme conditions, highlighting the practicality of our proposed scheme when applied to realistic scenarios.

The remainder of the paper is organized as follows: In Section II, we review the related works. In Section III, we present the system model. In Sections IV and V we formulate the problem. We discuss and analyze the simulation result in Section VI. Section VII concludes the paper.

II. Related Work

A. The Metaverse

In 1992, the term Metaverse first appeared in the science-fiction novel *Snow Crash* by Neal Stephenson [11]. The architecture of the Metaverse can be split into four layers: the physical world, the virtual world, the Metaverse engine, and the infrastructure [12]. The physical world contains the physical objects in the real-world, such as users, VSPs, and IoT sensors. The virtual world contains avatars, virtual environments, virtual goods/services, and tangible goods services. The Metaverse engine, such as VR/AR, AI, and blockchain, is used to obtain user input. Finally, the infrastructure ensures that the Metaverse is scalable, universally accessible, and user-reliable. The Metaverse is supported by various technologies, such as VR/AR, digital twin, and AI. To create a more effective digital twin in the Metaverse, continuous data synchronization is essential, involving real-time updates from the physical world to maintain the digital twin's accuracy. For example, a virtual driver training service requires ongoing data updates on road conditions, traffic, and weather for realistic training. IoT devices like autonomous cars, drones, and smartphones are considered for data collection in areas relevant to a virtual service provider (VSP) [13]. The data collected from the physical resources is the key to synchronizing the two worlds. The authors in [13] group drones with similar interests to speed up the data collection process and improve synchronization between the two worlds. The authors in [14] proposed to classify the resources in the Metaverse into three categories: cyber, physical, and people. In the Metaverse, it can be observed that many applications share common functions. For instance, in the Metaverse replica of Seoul (which will be launched by South Korea) [15], a digital map can serve as a shared feature between tourism and navigation applications. Therefore, the authors in [16] proposed a highly effective and comprehensive solution for managing and allocating resources by observing the common functions that can be shared.

B. Semantic Communication

Semantic communication was proposed as an intelligent communication scheme that focuses on the significance of transmitted messages rather than bitstream transmission [17]. The authors in [18] proposed a semantic communication framework in the wireless network, where a BS is used to transmit the semantic contents that are extracted from textual data to each user. A semantic similarity metric is used to measure the semantic accuracy and completeness of the recovered text. With the recent development of deep learning technologies, multiple semantic communication systems were developed for text transmission. The authors in [17] designed a deep learning-enabled semantic communication system known as DeepSC. DeepSC is used for semantic text transmission. With the help of Transformer, DeepSC seeks to maximize system capacity and reduce semantic errors when restoring the meaning of sentences. To understand the meaning in a sentence, *word2vec* model is introduced in [19] to capture the relationship between words. When the words are similar, their distance is closer in the vector space. *Word2vec* model can be used to extract the semantic information from sentences. However, *word2vec* model has to be redesigned whenever the context is different. Therefore, a general word representation model bidirectional encoder representations from transformers (BERT) is introduced in [20]. Depending on the context, BERT can generate more than one vector representation for the same word. Instead of developing encoding and decoding text or image transmission, we adopt semantic communication in the Metaverse context to reduce the number of transmissions and allocate the optimal edge sensing units to the VSPs.

C. Stochastic Optimization

There are several works that are working on deterministic optimization. The authors in [21] optimize the UAV trajectories to reduce the transmitted power required from the ground terminal. The authors in [22] used UAVs as a relay to deliver latency-critical messages with ultra-high reliability and then studied UAV location optimization and power allocation. However, the real world is full of uncertainties, and optimization should be done by considering the uncertainties. The authors in [14] used a two-stage stochastic integer programming (SIP) to allocate optimal resources to minimize the operation cost of the VSPs by considering the demand uncertainty of the users. The authors in [23] presented a data-driven risk-averse stochastic optimization framework and investigated the Wasserstein metric for continuous and discrete cases of uncertainty distribution. The authors in [24] considered task uncertainty and formulated a distributed robust optimization problem to minimize the expectation of system latency under the worst-case distribution.

III. System Model

We consider a Metaverse network shown in Figs. 1 and 2 shows the network topology. To develop the Metaverse, a set $\mathcal{W} = \{1, \ldots, w, \ldots, W\}$ of W VSPs subscribe to the semantic data provision plans offered by edge sensing units denoted by the set $\mathcal{E} = \{1, \ldots, e, \ldots, E\}$.

A. Semantic Market Data Provision Plans

The VSPs subscribe to the edge sensing units using two types of plans as follows:

- *Reservation plan:* This is a long-term plan in which the VSP w pays a *fixed* cost $c^r_{w,e}$ for n_e semantic data that has been previously collected and stored by edge sensing unit e. The fixed cost $c^r_{w,e}$ is dependent on the data collection, processing, and communication costs defined in Section III-B. As each edge sensing unit e has a different field of view at time slot $\bar{t}$, marked by coordinates $(x_e(\bar{t}), y_e(\bar{t}))$, the images that have been captured differ. The VSP chooses the reservation plan provided by an edge sensing unit that has the highest category similarity to its category interests, i.e., the VSP subscribes to the edge sensing unit that is able to transmit the semantic data that best satisfies its interest. Note that this is possibly a *many-to-one* matching between VSPs and the edge sensing unit. In other words, the edge sensing unit is able to sell its previously collected semantic data to as many interested VSPs as possible. The category similarity is further defined in Section III-C.
- *On-demand plan:* This is an ad-hoc plan triggered when the VSP requests for additional semantic data to be newly collected by the edge sensing unit e at time slot $\bar{t}$ for the on-demand *variable* cost $c^o_{w,e}(\lambda_i)$. λ_i denotes the demand scenario defined in (1). The ad-hoc plan is triggered when the semantic data from the reservation plan is insufficient to achieve a defined level of user QoE. We further define the QoE score in Section III-D. Unlike the reservation plan which is a fixed cost, the cost of the on-demand plan depends on the demand scenario i of all VSPs. Due to the unpredictability nature of such demands and *one-to-one* matching between VSP and edge sensing unit, more than one VSPs may compete for the on-demand sensing service of the edge sensing unit. As such, we devise a DRL-based double dutch auction as presented in Section IV to

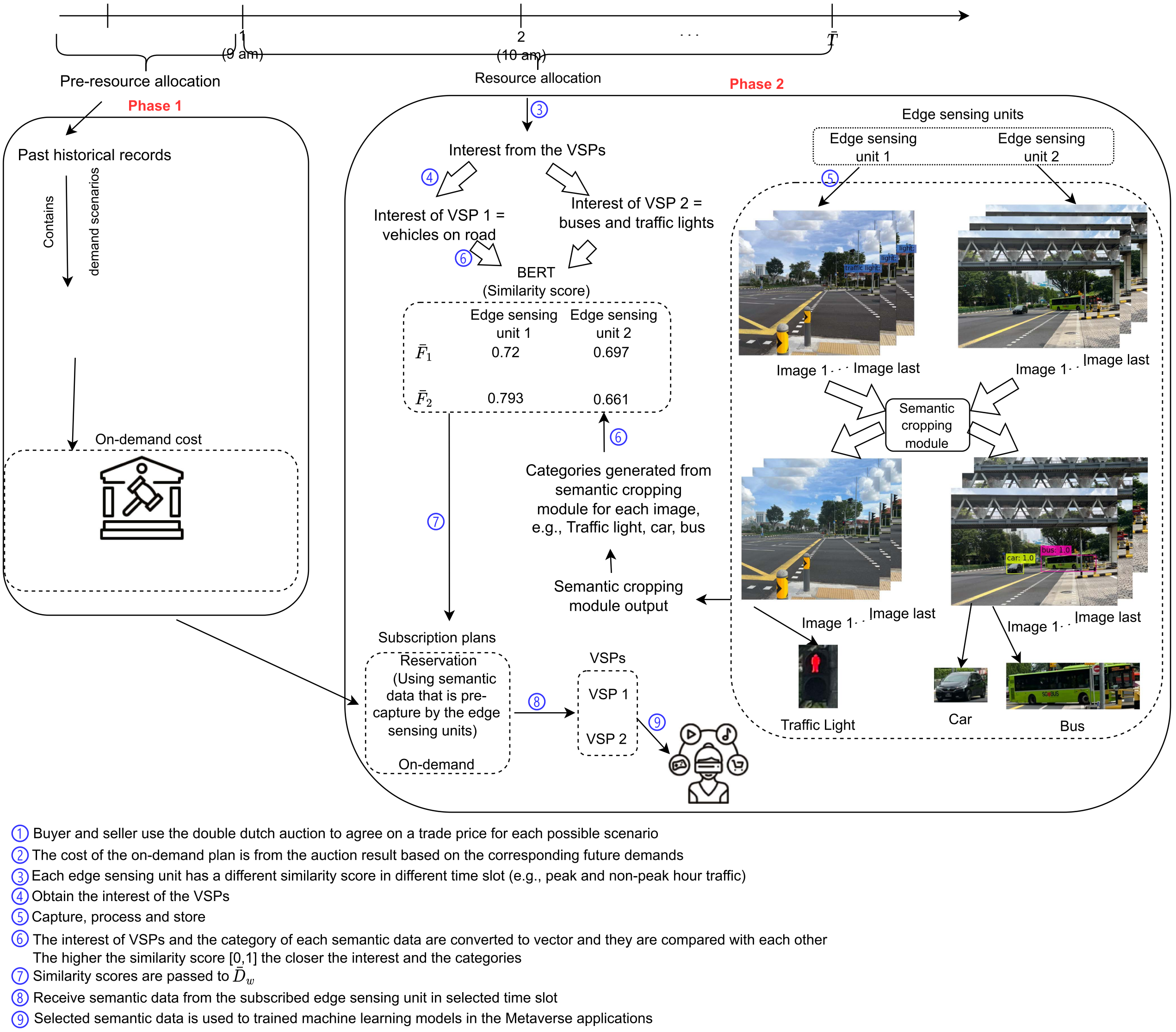

1
(9 am)
2
(10 am)
...
$\bar{T}$
Pre-resource allocation
Resource allocation
Phase 1
Phase 2
Past historical records
Contains
demand scenarios
On-demand cost
Interest from the VSPs
Interest of VSP 1 = vehicles on road
Interest of VSP 2 = buses and traffic lights
BERT (Similarity score)
Edge sensing unit 1
Edge sensing unit 2
$\bar{F}_1$ 0.72 0.697
$\bar{F}_2$ 0.793 0.661
Categories generated from semantic cropping module for each image, e.g., Traffic light, car, bus
Semantic cropping module output
Subscription plans
Reservation (Using semantic data that is pre-capture by the edge sensing units)
On-demand
VSPs
VSP 1
VSP 2
Edge sensing units
Edge sensing unit 1
Edge sensing unit 2
Image 1 · · · Image last
Semantic cropping module
Traffic Light
Car
Bus
3
4
5
6
7
8
9
1 Buyer and seller use the double dutch auction to agree on a trade price for each possible scenario
2 The cost of the on-demand plan is from the auction result based on the corresponding future demands
3 Each edge sensing unit has a different similarity score in different time slot (e.g., peak and non-peak hour traffic)
4 Obtain the interest of the VSPs
5 Capture, process and store
6 The interest of VSPs and the category of each semantic data are converted to vector and they are compared with each other The higher the similarity score [0,1] the closer the interest and the categories
7 Similarity scores are passed to $\bar{D}_w$
8 Receive semantic data from the subscribed edge sensing unit in selected time slot
9 Selected semantic data is used to trained machine learning models in the Metaverse applications

Fig. 1. Illustrative example of the network with two VSPs and two edge sensing units. A detailed illustration of the double dutch auction mechanism can be found in Fig. 4.

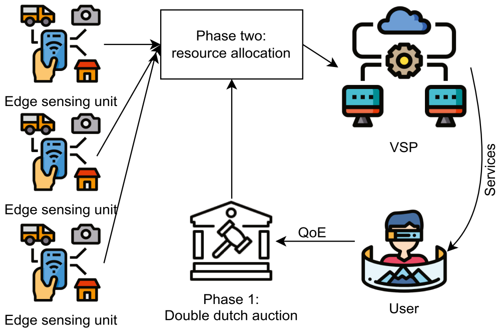


Fig. 2. Illustrative example of the network topology.

efficiently match the ad-hoc demand of the VSP to the limited supply of available edge sensing units.

In general, the reservation plan is cheaper than the on-demand plan in data markets [25], e.g., since additional data has to be collected on-demand within a short notice period. However, the on-demand plan has to be triggered due to scenario uncertainty. Let λ_i denote the i^{th} scenario of all the VSPs:

$$\lambda_i = (\bar{D}_1, \ldots, \bar{D}_W), \tag{1}$$

where $\bar{D}_w = (\gamma_{w,e}^{ave}, F_w, \bar{F}_w)$, $\gamma_{w,e}^{ave}$ is the average QoE score of the users who are accessing the Metaverse application developed by VSP w using semantic data from edge sensing unit e, $\bar{F}_W$ represents the set of category interest of VSP W, and F_W represents the number of semantic data transmissions that VSP W requires. Each scenario λ_i occurs with a probability of $P(\lambda_i)$.

B. *Sensing Model*

At its respective location, each edge sensing unit can sense (capture) images and process them to generate semantic data. The energy consumption for data collection can be expressed as follows [26]:

$$\bar{E}_e = (P_e^{idle}\hat{T}_e^{idle} + P_e^{active}\hat{T}_e^{active}), \tag{2}$$

where P_{idle} is the power consumption in idle state, P_{active} is the power consumption in active state, $\hat{T}_{idle}$ is the time in idle state, and $\hat{T}_{active}$ is the time in active state. $\hat{T}_{active} = \frac{d}{\tau_e}$, where d is the number of pixels in an image and τ_e denotes the total CPU computing capability of edge sensing unit e. When the captured data is processed locally, the local computation execution time of edge sensing unit e is expressed as [27]:

$$\hat{t}_e = \frac{\mathcal{C}_e d}{\tau_e}, \tag{3}$$

where $\mathcal{C}_e$ is the number of CPU cycles needed to process a bit.

When the semantic data is requested by the VSP, it is processed and transmitted through base station b which is located at (x_b, y_b, z_b). In an urban area, many obstacles surround the edge sensing units and prevent the units from maintaining a line of sight (LoS) transmission link with the BS. In order to identify whether a transmission link is LoS or Non-LoS (NLoS), the height of the obstacles has to be known, which is difficult in practice. Therefore, we adopt a probabilistic LoS channel model [28]. The LoS transmission probability from edge sensing unit e to BS b is defined as follows [29]:

$$P_{e,b}^L = B_3 + \frac{B_4}{1 + e^{-(B_1 + B_2\theta_{e,b})}}, \tag{4}$$

where $B_1 < 0$, $B_2 > 0$, $B_4 > 0$, and B_3 are the constants with $B_3 + B_4 = 1$. The probability of NLoS $P_{e,b}^N$ can be obtained as $1 - P_{e,b}^L$. $\theta_{e,b}$ is the elevation angle from e to b and is defined as:

$$\theta_{e,b} = \arctan\left(\frac{z_b}{d_{e,b}}\right), \tag{5}$$

where z_b is the height of BS b and $d_{e,b}$ is the horizontal distance between the edge sensing unit e and BS b. Then, the channel power gain between the edge sensing unit e and BS b can be modeled as:

$$h_{e,b} = P_{e,b}^L \beta_0 d_{e,b}^{-\alpha_L} + (1 - P_{e,b}^L)\mu\beta_0\, d_{e,b}^{-\alpha_N}, \tag{6}$$

where β_0 is the average channel power gain at a reference distance of 1 m, μ is the signal attenuation factor due to the NLoS propagation, α_L and α_N denote the average path loss exponents for the LoS and NLoS, respectively and $d_{e,b}$ is the distance between edge sensing unit e and BS b. To lighten the burden of the wireless network, we follow [30] to consider an orthogonal frequency-division multiple access (OFDMA) system to enable multiple concurrent uplinks and downlinks communication. Each orthogonal resource block can be occupied only by at most one transmission so that the interference caused by other edge sensing units can be avoided. The uplink bitrate for each edge sensing unit is given as follows:

$$r_e = B_e\left(\log_2\left(1 + \frac{\hat{P}_e h_{e,b}}{\sigma^2\Gamma}\right)\right), \tag{7}$$

where B_e denotes the allocated bandwidth of edge sensing unit e, $\hat{P}_e$ refers to the transmit power of edge sensing unit e, σ^2 denotes the receiver noise power and Γ denotes the signal-to-noise ratio (SNR) gap.

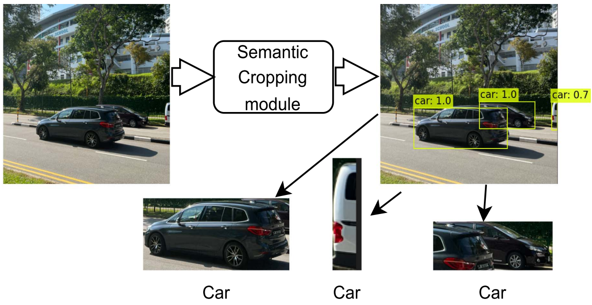


Fig. 3. Illustrative example of semantic cropping module (YOLO) output.

Accordingly, the reservation cost $c_{w,e}^r$ is defined as follows:

$$c_{w,e}^r = n_e\delta_1\left(\frac{\hat{P}_e\bar{q}}{r_e} + \bar{E}_e + \hat{t}_e\right) + \hat{F}_{w,e}, \tag{8}$$

where $\bar{q}$ is the average transmitted data size, δ_1 is the cost coefficient to convert energy consumption into cost, and $\hat{F}$ is the profit margin of the edge sensing unit, e.g., a fixed rental cost for its services. Similarly, the minimum on-demand cost (i.e., bid floor) is defined as follows to preserve the individual rationality of the auction:

$$\min(c_{w,e}^o) = \delta_2\left(\frac{\hat{P}_e\bar{q}}{r_e} + \bar{E}_e + \hat{t}_e\right), \tag{9}$$

where δ_2 is the cost coefficient and it should be chosen in such a way that satisfies $\frac{c_{w,e}^r}{n} \leq \min(c_{w,e}^o)$ as the cost of reservation plan $\leq$ the cost of the on-demand plan. The cost of the on-demand plan takes reference from the auction result by using the corresponding future demands. The actual on-demand cost is obtained from the double dutch auction output in Section IV.

C. Category Generation and Similarity Matching

Using the plans, the VSPs can obtain semantic data captured by the edge sensing units. However, it is difficult to identify which edge sensing unit produces images that are important or relevant to the interests of the VSPs. This paper adopts a pre-trained machine learning model, you only look once (YOLO) from [31]. YOLO provides real-time detection with relatively high accuracy. With the help of YOLO, objects (semantic data) and their corresponding categories can be extracted from the respective images. Fig. 3 shows an example of semantic cropping module (YOLO) output. When the category (label of the object) is within the interest of the VSPs, the semantic data, which is the snapshot of the object, can be transmitted.

However, once the categories are generated from the images, the VSPs cannot identify to which edge sensing unit to subscribe to as the VSPs do not know which semantic data is relevant to their interest. In additional, each image may output multiple semantic data and corresponding categories. It is not practical for the VSPs to search manually through the categories. Therefore, we propose to use category similarity for the VSPs to subscribe to the edge sensing unit that produces the best semantic data that meets the interest of the VSPs. Category similarity convert the interest of VSPs and the category of the semantic data into vectors so that they can be compared. However, in different contexts, the same word might have multiple definitions. For example, “wind” can mean the current of air or the action turn. The traditional method, such as *word2vec* cannot recognize polysemy. The issue arises when the same word cannot be represented by the same numerical vector in different contexts. One of the solutions is to use BERT [17]. BERT is a powerful pre-trained machine learning model that has been trained by billions of sentences for extracting semantic information. It is used to convert words into vectors according to different contexts. In this paper, we use cosine similarity [17] to measure the similarity between the sentences, i.e.,

$$\text{match}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}, \tag{10}$$

where $\mathbf{A}$ is the vectorized of $\bar{F}_w$ which is the interest of VSP w and $\mathbf{B}$ is the vectorized categories output generated from BERT. The category similarity defined in (10) is a number between 0 and 1, which indicates how similar $\mathbf{A}$ is to $\mathbf{B}$, with 1 representing the highest similarity and 0 representing no similarity. The average similarity of VSP w’s demands and edge sensing unit e’s data types is represented by $S_{w,e}$ and can be calculated from $S_{w,e} = \frac{\sum_{i=1}^{y_2} \sum_{j=1}^{y_1} \text{match}(\mathbf{A},\mathbf{B}_{ij})}{y_2}$, where y_1 is the total number of semantic data in an image and y_2 is the total number of images in edge sensing unit e. As a result, the VSPs can subscribe to receive the semantic data from the edge sensing unit that has the highest similarity score. $S_{w,e}$ prevents the VSP from over- and under-semantic data transmission. For example, the reservation plan from edge sensing unit 1 supports VSP 1 with 150 semantic data transmissions. After the interest and the demand (requires 300 semantic data, e.g., image segments) of VSP 1 is realized, the average similarity score $S_{1,1} = 0.8$. VSP 1 faces a shortfall of 180 as $300 - 0.8(150) = 180$. An additional on-demand plan is used to accommodate the shortfall of 225 semantic data, $\frac{180}{0.8}$. Therefore, a resource allocation scheme is required to prevent re-transmission and reduce the network’s energy consumption.

D. User Experience Metrics

VSPs provide users with immersive virtual environment experiences by using AR/VR devices. However, in the current virtual environment technology, some users exhibit cybersickness symptoms. There are numerous reasons that can contribute to cyber sicknesses, such as gender, age, illness, display, and technology issues [32]. Unfortunately, due to human complexity, it is difficult to tackle the cybersickness problem that is caused by individual factors. However, we can reduce the cybersickness effects if it is caused by display and technology, i.e., improve the rendering of the virtual environment. For example, when the lag occurs due to processing delays (insufficient data to render the 3D models), there is a delay between an individual’s action and the system’s reaction, and this can contribute to cybersickness symptoms [33]. By synthesizing images with neural networks, real-time rendering of 3D virtual environments can be achieved with low processing delays [34]. After the QoE from the users is known, more semantic data that are within the interest of VSPs can be requested from the edge sensing units to improve the QoE. In this paper, we use cybersickness to evaluate QoE [35]. Specifically, QoE can be quantified using parameters from the Simulator Sickness Questionnaire (SSQ) [36].

TABLE I
16 Symptoms of Cybersickness in SSQ [35]

SSQ Symptoms	N	O	D
General discomfort	1	1	
Fatigue		1	
Headache		1	
Eye strain		1	
Difficulty focusing		1	1
Increased salivation	1		
Sweating	1		
Nausea	1		1
Difficulty concentrating	1	1	
Fullness of head			1
Blurred vision		1	1
Dizzy (Eye open)			1
Dizzy (Eye closed)			1
Vertigo			1
Stomach awareness	1		
Burping	1		

The SSQ provides a comprehensive evaluation of a user’s experience by capturing various symptoms of simulator sickness, such as nausea, oculomotor issues, and disorientation. This allows for a more nuanced understanding of the user’s experience than other simpler metrics. Comparing with other metrics such as Weber-Fechner Law which represented by a logarithm function. SSQ directly captures the user’s subjective experience. This is crucial for Metaverse applications where the user’s comfort and well-being are primary concerns. As shown in Table I, SSQ consists of 16 parameters categorized into nausea (N), oculomotor (O), and disorientation (D). Each symptom is rated using a 4-point scale (0; no sickness, 1; mild sickness, 2; considerable sickness, 3; severe sickness). The score of each category is calculated by adding all the symptom ratings that fall under the same category. The total score is then measured by combining every single score for each category with the weight. The value of each SSQ symptom is between 0 and 4. The average SSQ score received by VSP w from its users for the nausea category can be calculated as [35]:

$$\gamma_w^{Nau} = \frac{\sum_{u=1}^{U_w} s_u^{gd} + s_u^{sw} + s_u^{is} + s_u^{dc} + s_u^{na} + s_u^{bu} + s_u^{sa}}{U_w}, \tag{11}$$

where s_u^{gd}, s_u^{sw}, s_u^{is}, s_u^{dc}, s_u^{na}, s_u^{bu}, and s_u^{sa} are the ratings for general discomfort, sweating, increased salivation, difficulty concentrating, nausea, burping, and stomach awareness, respectively. U_w represents the total number of users that use the Metaverse application provided by VSP w. The average oculomotor score received by VSP w from its users can be calculated as:

$$\gamma_w^{Ocu} = \frac{\sum_{u=1}^{U_w} s_u^{gd} + s_u^{he} + s_u^{fa} + s_u^{es} + s_u^{dc} + s_u^{df} + s_u^{bv}}{U_w}, \tag{12}$$

where s_u^{he}, s_u^{fa}, s_u^{es}, s_u^{df}, and s_u^{bv} are the rating for headache, fatigue, eyestrain, difficulty focusing, and blurred vision, respectively. The average disorientation score received by VSP w from its users can be calculated as:

$$\gamma_w^{Dis} = \frac{\sum_{u=1}^{U_w} s_u^{df} + s_u^{na} + s_u^{fh} + s_u^{bv} + s_u^{dzc} + s_u^{ve} + s_u^{dzo}}{U_w}, \tag{13}$$

where s_u^{fh}, s_u^{dzc}, s_u^{ve}, and s_u^{dzo} are the rating for the fullness of head, dizzy (eye close), vertigo, and dizzy (eye open), respectively.

To account for the QoE, the total average SSQ score is integrated with the semantic data that a VSP has:

$$\gamma_{w,e}^{ave} = \frac{1}{v_{w,e}}(v_1\gamma_w^{Nau} + v_2\gamma_w^{Ocu} + v_3\gamma_w^{Dis}), \tag{14}$$

$$\text{where } v_{w,e} = \log\left(1 + S_{w,e}F_w\right), \tag{15}$$

v_1, v_2, and v_3 are the weights for γ_w^{Nau} γ_w^{Ocu}, and γ_w^{Dis} respectively. Note that $v_{w,e}$ accounts for the relationship between the QoE of the users and the number of semantic data. When the SSQ (discomfort) score increases, the number of required semantic data also increases so that the VSPs can use more data in their applications to reduce the discomfort and improve the existing QoE. The relationship is constructed by using a logarithmic function following [37], [38].

E. Problem Formulation

To minimize the cost of the network, we introduce a two-phase optimization approach.

- *Phase one - On demand cost derivation:* For each scenario λ_i, the similarity score, SSQ score, and average transmitted semantic data size differs across the VSPs. Accordingly, the on-demand price fluctuates based on these factors, since the VSPs are facing different extents of shortfalls that affect their bidding decisions for the semantic data. We use the double dutch auction to obtain the distribution of on-demand cost paid by the VSPs to the owners of edge sensing units across the different scenarios.
- *Phase two - Distributed robust resource allocation:* In phase two, the VSPs will decide on the subscription plans to use, i.e., reservation or on-demand. The decision is made depending on the uncertainty of the demands and on-demand cost. Whenever there is insufficient data, the VSPs are allowed to purchase more semantic data with the use of an on-demand plan. We first formulate and optimize the problem using two-stage SIP. SIP uses historical data as the true distribution to model the uncertainties. However, the use of historical data to estimate real-world uncertainties is difficult and inaccurate. DRO is used to characterize the uncertainty of probability distribution of demands by constructing the confidence set, which is actually an alias of the uncertainty set and consists of a family of ambiguous probability distributions. With the help of observed historical data, which contains i) semantic interest from the VSPs, ii) the number of semantic data required, iii) QoE from the users who are accessing the Metaverse applications, iv) semantic data category from the edge sensing units, and v) the average transmitted semantic data size, the confidence set can be adjusted with a guaranteed confidence level. Then, resource allocation decisions can be made by using the constructed ambiguous probability distribution.

IV. Phase One: On-Demand Cost Derivation

In the pre-resource allocation stage, by using historical data, the VSPs and the edge sensing units can open a call market to place orders to buy or sell semantic data at certain bid prices and match at predetermined time intervals. The VSPs will buy semantic data transmission from the owners of the edge sensing units in this call market. Therefore, the VSPs are the buyer, owners of the edge sensing units are the sellers, and they can be supervised by an auctioneer such as the network operator or administrator. We design a DRL-based double dutch auction mechanism to determine the buying price from the VSPs and the selling prices from the owners of the edge sensing units. The price that both the seller and buyer agree with each other is the on-demand cost. The detailed explanation of the different roles in the call market can be found below:

- *Buyer:* At the start of the auction, the buyer will submit their buy-bid to the Metaverse administrator. The buy-bid of VSP w at each time step t is denoted as $k_{w,e}^t$. This is the maximum price that VSP w is willing to pay to the edge sensing unit, and it is defined as follows [39]:

$$k_{w,e}^t = \delta_0\gamma_{w,e}^{ave}, \tag{16}$$

where δ_0 is the cost coefficient. When the $\gamma_{w,e}^{ave}$ score is high, it means that the users are faced with severe simulator sickness and have low QoE. Therefore, VSP w is willing to pay more to improve its application.
- *Seller:* Since the edge sensing unit is providing the communication resource to the buyer, the selling bid of the edge sensing unit should depend on the transmission energy consumption. This is the minimum price that the seller is willing to sell to prevent a loss in profit. At the start of the auction, the seller submits their sell-bid o_e^t to the Metaverse administrator, and it is defined as follows [39]:

$$o_e^t = \frac{\hat{P}_e\bar{q}\delta_1}{r_e}, \tag{17}$$

where $\bar{q}$ is the average transmitted data size and δ_1 is the cost coefficient with a similar role to δ_0.
- *Auctioneer:* The auctioneer manages clocks, denoted as $C_{B,w,e}^t$ and $C_{S,e}^t$ for the buyer side and the seller side, respectively. $C_{B,w,e}^t$ is the buying clock that displays the buying price for buyer w to seller e. It descends over time and displays the buying price at each step. $C_{S,e}^t$ is the selling clock, and it ascends and displays the selling price for the seller e at each step. At the beginning of the auction, $C_{B,w,e}^0$ opens at the highest price p^{max}, and $C_{S,e}^0$ opens at the lowest price p^{min}.

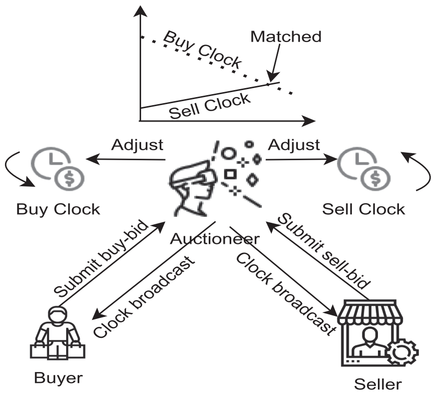


Fig. 4. Illustrative example of the double dutch auction mechanism.

A. Double Dutch Auction Mechanism Actions

An illustrative example of the double dutch auction mechanism is shown in Fig. 4. Generally, the buyers in a double dutch auction begin to count at a very high price and continue to count downward. In contrast, the sellers begin to ascend from a very low price and continue to ascend until a seller stops it (who offers the product at that price). When the buyer and seller do not match at the trade price, the buyer clock resumes its downward movement, followed by the seller clock's upward movement. Finally, the trading is over when the two prices cross (purchase is made at the crossover point) [40], [41]. In the double dutch auction mechanism, there are mainly three types of actions, i.e., clock broadcast, clock acceptance, and clock adjustment.

- *Clock broadcast:* The double dutch mechanism begins with a clock broadcast. The auctioneer will broadcast the current buying and selling clocks to the buyers or the sellers at the start of each iteration.
- *Clock acceptance:* After receiving the auction clock from the Metaverse administrator, the buyers or the sellers will compare their bids with the current clocks. The buyers or the sellers will bid if the current clock meets the buy-bid or sell-bid conditions. For the buyers, the w^{th} buyer who is not in $\mathcal{M}_B^t$ can submit its buy-bid $b_{w,e}^t$, which represents the maximum price that a buyer is willing to pay and $b_w^t = C_{B,w,e}^t$. $\mathcal{M}_B^t$ is a set that contains all the buyers who are matched with a seller, and $\mathcal{M}_S^t$ is a set that contains all the sellers who are matched with a buyer. Similar to the buyers, the sellers who have not participated in the trade will decide after receiving the broadcasted selling clock from the Metaverse administrator. The sellers can submit their sell-bid a_e^t, which represents the minimum price that a seller is willing to sell, and $a_e^t = C_{S,e}^t$. The buyer and seller are matched with each other if $k_{w,e}^t \geq C_{B,w,e}^t$, $o_e^t \leq C_{S,e}^t$, and the buy clock crosses the sell clock at $t+1$, i.e., $C_{S,e}^{t+1} \geq C_{B,w,e}^{t+1}$. Then, $\mathcal{M}_B^t = \mathcal{M}_B^t \cup \{w\}$, $\mathcal{M}_S^t = \mathcal{M}_S^t \cup \{e\}$, and $\mathcal{M}_{price}^t = \mathcal{M}_B^t \cup \{C_{w,e}\}$. $\mathcal{M}_{price}^t$ is a set that contains the final matched price and $C_{w,e}$ is defined as follows:
$$C_{w,e} = \frac{C_{B,w,e}^t + C_{S,e}^t}{2}. \tag{18}$$
Thus, the difference between the expected buy-bid and the real buy-bid is the regret $g_{w,e}^t$ of buyer w, which can be calculated as:
$$g_{w,e}^t = k_{w,e}^t - b_{w,e}^t. \tag{19}$$
On the other hand, the difference between the expected sell-bid and the real sell-bid is the regret g_e^{t+1} of seller e, which can be calculated as:
$$g_e^t = a_w^t - o_e^t. \tag{20}$$
- *Clock adjustment:* Clock acceptance only occurs when the sell-bid or buy-bid conditions are met. Most of the time, it will be clock adjustment, which means that none of the sellers or the buyers is participating in the auction, so the auction clocks must be adjusted accordingly. For example, when the buy-bid condition is not met, the Metaverse administrator chooses a step size of μ^t to adjust the buyer clock as follows:
$$C_{B,w,e}^{t+1} = C_{B,w,e}^t - \mu^t. \tag{21}$$
On the other hand, if the sell-bid condition is not met, the Metaverse administrator chooses a step size of μ^t to adjust the seller clock:
$$C_{S,e}^{t+1} = C_{S,e}^t - \mu^t. \tag{22}$$
The Metaverse administrator can dynamically adjust the step size to improve the auction's efficiency via proper step size. The auction ends when all sellers' and buyers' clocks cross with each other, i.e., $C_{B,w,e}^{T+1} < C_{S,e}^{T+1}$.

B. Markov Decision Process of the Double Dutch Auction

In order to determine the proper step size and minimizes the regrets of the buyers and sellers. The double dutch auction mechanism can be formulated as a Markov Decision Process and is defined as $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} >$, where $\mathcal{S}, \mathcal{A}, \mathcal{P}$, and $\mathcal{R}$ are the state space, action space, state transition probability function, and reward, respectively.

- *State space:* The state space at the current time step t contains the buyer clock $C_{B,w,e}^t$, the seller clock $C_{S,e}^t$, the set of winning buyers $\mathcal{M}_B^t$, the set of winning sellers $\mathcal{M}_S^t$, the final matched price $\mathcal{M}_{price}^t$, and the total state space is defined as follows:
$$\mathcal{S}^t = \{C_{B,w,e}^t, C_{S,e}^t, \mathcal{M}_B^t, \mathcal{M}_S^t\} \tag{23}$$
- *Action space:* The action space contains the step μ^t that the Metaverse administrator used to adjust the buyer and the seller clock at time step t, i.e., $a^t = \{\mu^t\}$.
- *State transition:* The state transition of the Metaverse administrator from the current state $\mathcal{S}^t$ to the next state $\mathcal{S}^{t+1}$ depends on the action space a^t.
- *Reward:* Following [39], the reward consists of two parts, the trade regret of buyers and sellers and the auction information exchange cost to broadcast the new auction clock to buyers or sellers. The reward is shown in Algorithm 1.

Algorithm 1: Algorithm for Calculating the Reward in DRL.

```
if $k^t_{w,e} \geq C^t_{B,w,e}$ then
    if $o^t_e \leq C^t_{S,e}$ then
        if Buy clock crosses the sell clock then
            $r(\mathcal{S}^t, a^t) = -g^t_{w,e} - g^t_e$
else
    $r(\mathcal{S}^t, a^t) = -p_a(W) - p_a(E)$
```

Note that $p_a(X)$ is the auction information exchange cost function for X participants and $p_a(X) = cX$, where c is denoted as the cost coefficient.

To maximize the long-term accumulated reward, the Metaverse administrator needs to determine its optimal action $a \in \mathcal{A}$ given state $s \in \mathcal{S}$. The optimal policy determined by the Metaverse administrator is defined as $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$. The Q-learning can be adopted to find the optimal policy by constructing a look-up table with Q-values of state-action pairs, i.e., $Q(s, a)$. The Q-values are updated based on the experiences of the Metaverse administrator as follows:

$$Q^{new}(s, a) = (1 - \Lambda)Q(s, a) + \Lambda\bigg(r(s, a) + \gamma \max_{a' \in \mathcal{A}} Q(s', a')\bigg), \quad (24)$$

where Λ is the learning rate, γ is the discount factor, and $r(s, a)$ is the reward received. However, Q-learning has the issue of large state and action spaces. Thus, in this paper, we use deep Q-learning (DQL) to find the optimal policy to obtain the on-demand cost which is the selling price of the VSP.

C. Deep Q-Learning

DQL uses a neural network to approximate the action-value function $Q(s^t, a^t)$, so the Q-value at time step t can be rewritten as $Q(s^t, a^t|\theta)$ and θ is the parameter of the neural network. After the approximation, the optimal policy $\pi^*(s)$ will be given by [42]:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s^t, a^t|\theta), \quad (25)$$

where $Q^*(s_t, a_t|\theta)$ is the optimal Q-value via DNN approximation. DQN will choose the approximated action $a^{t+1} = \pi^*(s^{t+1})$. Then, the approximated $\tilde{Q}(s^t, a^t|\theta)$ can be formulated as:

$$\tilde{Q}(s^t, a^t|\theta) = r(\mathcal{S}^t = s^t, a^t = a^t) + \gamma Q(s^{t+1}, \pi^*(s^{t+1}|\theta)). \quad (26)$$

The parameter θ is updated by minimizing the loss, defined as follows:

$$L = \frac{1}{T}\sum_{t=0}^{T}\bigg(\tilde{Q}(s^t, a^t|\theta) - Q(s^{t+1}, \pi^*(s^{t+1}|\theta))\bigg)^2. \quad (27)$$

At time instance t, the Metaverse administrator will choose action a^t according to (25), get a reward $r(\mathcal{S}^t, a^t)$ and proceed to the next state s^{t+1}. The vector (s^t, a^t, r^t, s^{t+1}) will be stored in experience memory. Then the parameter θ and policy π will be updated according to the data fetched from the experience memory [42]. By using the DRL-double dutch auction mechanism, a trade list Γ_i is generated for each demand scenario λ_i. The total number of trade lists generated is equal to the number of demand scenarios. The trade list contains the matched price and the matched buyer and seller. The on-demand cost from phase two uses the matched price to perform the resource allocation whenever the scenario occurs.

We analyze the complexity of the proposed DRL-enabled double dutch auction. Following [43], the complexity of the proposed DRL is $O(\hat{e}|\theta|\hat{C})$. $\hat{e}$ is the number of epochs, $|\theta|$ is the number of parameters in the neural network. $\hat{C}$ is the time complexity of calculating the gradient of each θ.

D. Economic Properties

This section discusses the desirable properties of the double dutch auction, i.e., individual rationality, truthfulness, and budget balance.

Theorem 1: The proposed double dutch auction satisfies individual rationality.

Proof: Individual rationality is ensured when none of the participating parties loses any money when the auction is closed. This means that the utilities for both the sellers and buyers are non-negative. A buyer may face two conditions, i) the buyer wins the auction, and ii) the buyer loses the auction. When a buyer wins the auction, the utility of a buyer w can be expressed as $\bar{u}_w = b^t_{w,e} - C_{w,e}$. Since $b^t_w = C^t_{B,w,e}$ and $C^t_{B,w,e} > C^t_{S,e}$, it leads to $b^t_{w,e} > C_{w,e}$. Therefore, in this case, the utility of a buyer w is non-negative. However, in the latter situation, a user's utility is 0 if the user loses an auction. We may generally state that buyers never experience any losses.

On the other hand, the seller may also experience the same two situations. When a seller wins the auction, the utility of a seller e can be expressed as $\bar{u}_e = C_{w,e} - a^t_e$. Since $a^t_e = C^t_{S,e}$ and $C^t_{B,w,e} > C^t_{S,e}$, it leads to $C_{w,e} > b^t_{w,e}$. Therefore, in this case, the utility of a seller e is non-negative. In the second situation, the seller who loses an auction neither earns nor loses anything. Therefore, the proposed auction satisfies individual rationality.

Theorem 2: Proposed double dutch auction satisfies truthfulness:

Proof: Truthful auctions stop malevolent users from placing a fraudulent bids. The payment of buyers' and sellers' revenue is determined by the common crossing price $C_{w,e}$. Therefore, buyers and sellers do not have any other incentive to submit their buy-bids or sell-bids except their true valuations to the semantic transmissions.

Theorem 3: Proposed double dutch auction is budget balance.

Proof: When the auctioneer makes no losses, an auction is said to be budget balanced [41]. Since the buyers and sellers trade with the common crossing price $C_{w,e}$, their utility is non-negative. The auctioneer gains a non-negative utility as the auctioneer received $p_a(W) + p_a(E)$ whenever the auction information is exchanged.

V. Phase Two: Distributed Robust Resource Allocation

Once the trade list Γ_i is obtained for each scenario λ_i. The cost of the on-demand plan will be equal to the matched price in Γ_i. In the following section, we first provide a detailed stochastic integer programming formulation, and then we extend the formulation to distributional robust optimization to allocate the semantic resources in the Metaverse.

A. Stochastic Integer Programming Formulation

In this section, we develop a two-stage Stochastic Integer Programming (SIP) to minimize the total cost of the network by optimizing the subscription plans. The two-stage SIP can be solved by assuming that the probability distribution of the demand is known. The first stage defines which edge sensing unit is subscribed to the reservation plan while the second stage defines the number of semantic data by utilizing the on-demand plan. We define two variables to indicate the subscription plan that is used. $m^{\mathrm{r}}_{w,e}$ is a binary variable indicates whether VSP w want to rent edge sensing unit e using the reservation plan or not. For example, $m^{\mathrm{r}}_{w,e} = 1$ means that the VSP rents edge sensing unit e and $m^{\mathrm{r}}_{w,e} = 0$ means otherwise. $m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i)$ is a positive variable to indicate the number of semantic data to purchase by VSP w from edge sensing unit e in time slot $\bar{t}$. Let a convex sample space Ω contain I possible scenarios. The two-stage SIP formulation can be expressed as follows:

$$\min_{m^{\mathrm{r}}_{w,e},\, m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i)} :$$

$$\sum_{w\in\mathcal{W}}\sum_{e\in\mathcal{E}} m^{\mathrm{r}}_{w,e} c^{\mathrm{r}}_{w,e} + \sum_{\lambda_i\in\Omega} P(\lambda_i) \sum_{\bar{t}\in\bar{\mathcal{T}}}\sum_{w\in\mathcal{W}}\sum_{e\in\mathcal{E}} m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i) c^{\mathrm{o}}_{w,e}(\lambda_i)\delta_2, \tag{28}$$

subject to:

$$m^{\mathrm{r}}_{w,e} \geq \bar{O} m^{\mathrm{r}}_{\bar{t},w,e}, \qquad \forall \bar{t}\in\bar{\mathcal{T}}, \forall w\in\mathcal{W}, \forall e\in\mathcal{E}, \tag{29}$$

$$\sum_{\bar{t}\in\bar{\mathcal{T}}}\sum_{e\in\mathcal{E}} m^{\mathrm{r}}_{\bar{t},w,e} n \bar{F}_{\bar{t},w,e}(\lambda_i) + \sum_{\bar{t}\in\bar{\mathcal{T}}}\sum_{e\in\mathcal{E}} m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i)\bar{F}_{\bar{t},w,e}(\lambda_i) \geq F_w, \forall w\in\mathcal{W}, \forall\lambda_i\in\Omega, \tag{30}$$

$$\sum_{w\in\mathcal{W}} m^{\mathrm{r}}_{w,e} \leq 1, \qquad \forall e\in\mathcal{E}, \tag{31}$$

$$m^{\mathrm{r}}_{w,e} \in \{0,1\}, \qquad \forall w\in\mathcal{W}, \forall e\in\mathcal{E}, \tag{32}$$

$$m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i) \in \mathbb{Z}^+, \quad \forall \bar{t}\in\bar{\mathcal{T}}, \forall w\in\mathcal{W}, \forall e\in\mathcal{E}, \forall\lambda_i\in\Omega. \tag{33}$$

$\bar{O}$ indicates a small number and $m^{\mathrm{r}}_{\bar{t},w,e}$ is a decision variable. δ_2 is the weight to ensure the on-demand cost is within an acceptable range. Constraint (29) ensures that the semantic transmission is available for VSPs for all the time slots whenever an edge sensing unit is rented. At each time slot, a single edge sensing unit can support n semantic data transmission and $\bar{F}_{\bar{t},w,e}(\lambda_i)$ is the similarity score between the semantic data and the VSP interest. (30) ensures that the demand of the VSP has to be met using the on-demand plan whenever there is insufficient semantic data from the reservation plan. (31) ensures that each edge sensing unit e can only be used once in the reservation plan. We utilize the GAMS script to output the respective decision variables and use the number of constraints to analyze the complexity of the two-stage SIP [10], [44]. Let $\bar{t}'$, w', e', and λ' denote the total number of time slots, VSPs, edge sensing units, and demand scenarios. Then, the total number of constraints are $w'\lambda' + w'e'(1 + \bar{t} + \lambda')$.

B. DRO Cost Minimization

In the real-world Metaverse scenario, the uncertainties generated by the users are highly dynamic and the probability distribution of demand is unknown and hard to predict accurately. Inherent uncertainty and inaccurate estimation of the probability distribution seriously affect the network cost. Therefore, this section introduces DRO to minimize the expected total cost under the worst-case distribution realization in D, where D represents the confidence set for the ambiguous distribution, constructed in Section V-B1. Different from SIP, which assumes that the probability distribution of demand is known. DRO construct a confidence set and presumed that the confidence set contains the true distribution because it is data-driven.

1) Confidence Set Construction: In this paper, there is no clear probability distribution for the demand and overall average SSQ score that the VSPs received. As an alternative, a confidence set for ambiguous distributions is created by explicitly associating the model's robustness when applied to the accumulated historical data.

The reference distribution is expressed as $\mathcal{P}_0 = \{p^0_1, p^0_2, \ldots, p^0_{I'}\}$ for a series of obtained historical data with a total of I' scenarios. Note that p^0_1 and $P(\lambda_i)$ are interchangeable. We use the empirical distribution as the reference distribution [24]. Let $\mathcal{P} = \{p_1, \ldots, p_I\}$ denote the ambiguity/true probability distribution. Taking into account that the reference distribution $\mathcal{P}^0$ is not always identical to the true distribution $\mathcal{P}$, we calculate the distance between them and we create a confidence set for the ambiguous distribution, denoted as D. However, a different metric used will impact the effectiveness of the system differently. Therefore, we adopt L_1 and L_∞ norm because of their excellent numerical tractability. L_∞ norm is lesser conservative than L_1 norm. This means that L_∞ norm is willing to take on more risk, is less focused on worst-case scenarios, or has a tighter confidence set than the L_1 norm. As a result, L_1 norm is more reliable at the expense of higher cost [24]. The distance measurement for the L_1 and L_∞ norm are denoted as $d_1(\mathcal{P}, \mathcal{P}_0)$ and $d_\infty(\mathcal{P}, \mathcal{P}_0)$ and the confidence sets D_1 and D_∞, which are corresponding to L_1 and L_∞ norm, respectively, can be constructed as follows [24]:

$$d_1(\mathcal{P}, \mathcal{P}_0) = \sum_{i=1}^{I} |p_i - p^0_i|, \tag{34}$$

$$d_\infty(\mathcal{P}, \mathcal{P}_0) = \max_{1\geq i\geq I} |p_i - p^0_i|. \tag{35}$$

$$D_1 = \left\{ \mathcal{P} \in \mathcal{R}_+^I \middle| \sum_{i=1}^{I} |p_i - p_i^0| \leq \theta_1 \right\}, \tag{36}$$

$$D_\infty = \left\{ \mathcal{P} \in \mathcal{R}_+^I \middle| \max_{1 \geq i \geq I} |p_i - p_i^0| \leq \theta_\infty \right\}, \tag{37}$$

where θ_1 and θ_∞ are the tolerance values, and they are closely related to both the confidence level β_1, β_∞, and the size of available historical data I'. Their exact relationship can be determined from the convergence rates between the ambiguous distribution $\mathcal{P}$ and the reference distribution $\mathcal{P}^0$ under the L_1 and L_∞ norm [24]. The convergence rate are defined as follows [24]:

$$Pr\{||\mathcal{P} - \mathcal{P}^0||_1 \leq \theta_1\} \geq 1 - 2I \exp\left(\frac{-2I'\theta_1}{I} \right), \tag{38}$$

$$Pr\{||\mathcal{P} - \mathcal{P}^0||_\infty \leq \theta_\infty\} \geq 1 - 2I \exp(-2I'\theta_\infty). \tag{39}$$

Suppose $\beta_1 = 1 - 2I \exp(\frac{-2I'\theta_1}{I})$ and $\beta_\infty = 1 - 2I \exp (-2I'\theta_\infty)$, $\theta_1 = -\frac{I}{2I'} \ln \frac{1-\beta_1}{2I}$ for L_1 norm metric and $\theta_\infty = -\frac{1}{2I'} \ln \frac{1-\beta_\infty}{2I}$ for L_∞ norm metric [24]. This means that the probability of the ambiguous distribution within the confidence sets that are constructed by L_1 and L_∞ norms are at least equal to β_1 and β_∞ respectively.

2) DRO Problem Formulation: Based on the distributed robust optimization model and using the two-stage SIP from (28), we formulate our optimization problem, the objective of which is to minimize the expected network cost under the worst-case distribution realization in D. Here, D can be constructed according to the L_1 and L_∞ norms, which are corresponding to D_1 and D_∞. Then, the formulated problem is written as:

$$\min_{m^{\mathrm{r}}_{w,e}, m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i)} \max_{\mathcal{P} \in D} : \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} m^{\mathrm{r}}_{w,e} c^{\mathrm{r}}_{w,e} + \sum_{\lambda_i \in \Omega} P(\lambda_i) \sum_{\bar{t} \in \bar{\mathcal{T}}} \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i) c^{\mathrm{o}}_{w,e}(\lambda_i) \delta_2, \tag{40}$$

subject to: (29)–(33).

We solve the optimization by first interchanging the minimization and maximization operations in the objective function (28). Then, the optimization problem can be rewritten as follows:

$$\max_{\mathcal{P} \in D} \min_{m^{\mathrm{r}}_{w,e}, m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i)} : \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} m^{\mathrm{r}}_{w,e} c^{\mathrm{r}}_{w,e} + \sum_{\lambda_i \in \Omega} P(\lambda_i) \sum_{\bar{t} \in \bar{\mathcal{T}}} \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} m^{\mathrm{o}}_{\bar{t},w,e}(\lambda_i) c^{\mathrm{o}}_{w,e}(\lambda_i) \delta_2, \tag{41}$$

subject to: (29)–(33).

To address the problem in (41), we can treat the probability distribution P as fixed and then optimize the internal minimization problem. However, there is still a maximization in the outer layer in (41). Thus, we turn the minimization problem into its dual problem, which is illustrated as follows:

$$\max_{y^{(1)}_{\bar{t},w,e}, y^{(2)}_{\bar{t},w,e}(\lambda_i), y^{(3)}_{w,e}} : \sum_{\bar{t} \in \bar{\mathcal{T}}} \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} \sum_{\lambda_i \in \Omega} y^{(2)}_{\bar{t},w,e}(\lambda_i) F_w(\lambda_i) - \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} y^{(3)}_{w,e}, \tag{42}$$

subject to:

$$- y^{(1)}_{\bar{t},w,e} - y^{(3)}_{w,e} \leq 0, \forall \bar{t} \in \bar{\mathcal{T}}, \qquad \forall w \in \mathcal{W}, \forall e \in \mathcal{E}, \tag{43}$$

$$\sum_{\bar{t} \in \bar{\mathcal{T}}} \sum_{e \in \mathcal{E}} \left(\bar{O} y^{(1)}_{\bar{t},w,e} + n \bar{F}_{\bar{t},w,e}(\lambda_i) y^{(2)}_{\bar{t},w,e}(\lambda_i) \right) \leq \sum_{e \in \mathcal{E}} C^{o}_{w,e}(\lambda_i) P(\lambda_i) \delta_2, \qquad \forall w \in \mathcal{W}, \forall \lambda_i \in \Omega, \tag{44}$$

$$\sum_{w \in \mathcal{W}} y^{(3)}_{w,e} \leq c^{r}_{w,e}, \qquad \forall e \in \mathcal{E}, \tag{45}$$

where $y^{(1)}_{\bar{t},w,e}$, $y^{(2)}_{\bar{t},w,e}(\lambda_i)$, and $y^{(3)}_{w,e}$ are dual variables corresponding to constraints (29)–(31).

As a result, the dual problem and the outer maximization operation can be combined and the final reformulated form of the original optimization problem is illustrated as follows:

$$\max_{P(\lambda_i), y^{1}_{\bar{t},w,e}, y^{(2)}_{\bar{t},w,e}(\lambda_i), y^{(3)}_{w,e}} : \sum_{\bar{t} \in \bar{\mathcal{T}}} \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} \sum_{\lambda_i \in \Omega} y^{(2)}_{\bar{t},w,e}(\lambda_i) F_w(\lambda_i) - \sum_{w \in \mathcal{W}} \sum_{e \in \mathcal{E}} y^{(3)}_{w,e}, \tag{46}$$

subject to: (43)–(45)

$$\sum_{\lambda_i \in \Omega} P(\lambda_i) = 1, \tag{47}$$

$$\mathcal{P} \in D, \tag{48}$$

where (47) is the basic condition of probability distribution, and (48) indicates that the ambiguous distribution $\mathcal{P}$ belongs to the confidence set D. The DRO algorithm can be solved in two steps. In step 1, given historical data and using it as a reference distribution $\mathcal{P}_0$, the problem (46) can be solved using the GAMS script to construct the ambiguous distribution $\mathcal{P}$. In step 2, by substituting $\mathcal{P}$ into the objective function of inner minimization problem (41), it is a linear programming problem, and we also utilize the GAMS script to output the respective decision variables. Similar to Section V-A, we also analyze the complexity of the DRO by using the number of constraints. The total number of constraints in both step 1 and step 2 of DRO are $2\,w'\lambda' + e' + 1 + N_{metric} + w'e'(1 + 2\bar{t} + \lambda')$. N_{metric} is the number of linear inequality and equality constraints introduced by the adopted metric [24].

VI. Performance Evaluation

We consider three virtual service providers, and three edge sensing units i.e., smartphones (model: iPhone 13 Pro Max), e_1, e_2, and e_3. The interest of the VSPs are "vehicles on

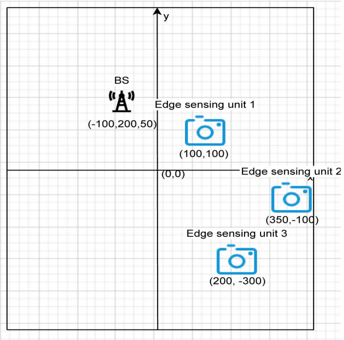


Fig. 5. x-y coordinates of edge sensing units and base station (BS).

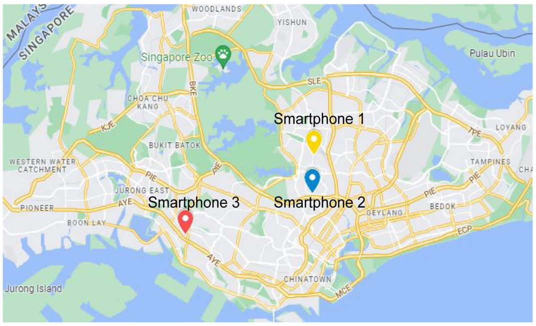


Fig. 6. Locations of the edge sensing units, such as smartphones that are used to collect data around Singapore.

TABLE II
EXPERIMENT PARAMETERS [29]

Parameter	Values
α_L	2.5
α_N	3.5
B_e	[0.1,0.2] Mhz
B_1	-0.4568
B_2	0.0470
B_3	-0.63
B_4	1.63
μ	-20dB
β_0	-60 dB
Γ	8.2 dB
σ^2	-109 dBm
Uplink transmission power,$\hat{P}_e$	20 dBm

road", "buses and traffic lights" and "motorcyclist and motorcycle". Edge sensing devices are deployed in a rectangular area 800 × 1000 m with three points as vertices where the locations are randomly generated as shown in Fig. 5, (100, 100) m, (350, -100) m, and (200, -300) m. The location of the BS is at (-100, 200, 50) m. Fig. 6 shows the locations where the data are collected by the edge sensing units deployed around Singapore. Fig. 7 shows the examples of data that are collected from smartphone 1 and Fig. 8 shows the examples of semantic data that are processed. Each smartphone's daily rental cost $\hat{F}$ is \$1.89 [45]. The other system parameters are shown in Table II. In the simulations, we have generated the SSQ behaviors randomly to determine $\gamma_{w,e}^{ave}$. At the same time, F_W is also generated randomly in each demand scenario.

3) Energy Efficiency: We first compare the energy efficiency between semantic data transmission and non-semantic data transmission. In semantic transmission, the energy consumption components consist of data collection, semantic data processing (deep learning model inference), and semantic data transmission. On the other hand, the energy consumption components of non-semantic transmission consist of data collection and data transmission. The energy consumption for data collection is negligible as it exists in both. The most intensive operation in semantic data processing is YOLO object detection. According to [46], the energy usage of YOLO is around 4.29×10^{-4} J, which is 4.7879×10^{-4} times less than semantic data transmission. If the energy consuming of YOLO is added with the energy consumption of semantic data transmission, the final result is not affected after we round off to two decimal places. Therefore, the energy consumption for the data process is negligible. An image's average transmitted data size is 2.2Mb, while the average transmitted semantic data is 45Kb. The reason is that the transmitted semantic data is precisely the region in which the VSP is interested (demand). Hence, the energy of semantic data transmission is 0.9 J, and the energy of non-semantic data transmission is 44 J. The energy usage for semantic data processing is 2.232 J/image [47]. The total energy consumption is shown in Fig. 9. With the help of semantic data communication, edge sensing units can reduce their power consumption during transmission as well as storage costs, which means that they will charge the VSPs less as the transmission cost depends on the transmission energy. In addition, it improves the sustainability of developments in the Metaverse.

A. *Convergence of Double Dutch Auction*

As shown in Fig. 10, the reward of the DQL in Section IV-A can be maximized and converged at around 3500 epochs. In DQL, we use the epsilon-greedy policy to select an action to balance exploration and exploitation. Exploration allows an agent to increase its existing understanding of each action, resulting in a long-term benefit. On the other hand, exploitation is taking random actions to collect unknown information. In an optimal long-term approach, it may involve some short-term sacrifices. For example, one exploratory attempt will lead to the lowest reward, but it is necessary to serve as an experience to avoid doing so in the future. At around 700 and 3000 epochs, the DQL is making random actions that lead to poor results and reduce the overall reward.

B. *Varying Time Steps*

A simulation setup is used to illustrate the DRL-double dutch auction result and it is shown in Fig. 11. This simple network consists of two VSPs and two edge sensing units. In this simulation, the buyer clock reduces from $k_{w,e}^t$, and the seller clock increases from o_e^t. The auction terminates when all the seller clocks cross the buyer clocks. Using the match condition from Section IV-A, seller 2 is matched with buyer 2 at $t = 6$, and therefore the matched price is determined by using the clock value from $t - 1$. Once buyer 2 and seller 2 are matched, they are removed from the auction. Similarly, for buyer 1 and seller 1,

Fig. 7. Examples of data that are collected from smartphone 1.

Fig. 8. Examples of semantic data that are processed by smartphone 1.

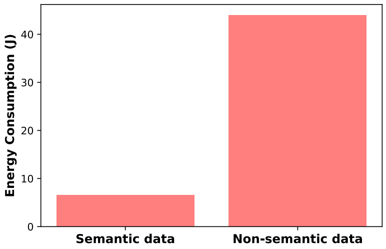


Fig. 9. Energy consumption of semantic data and non-semantic data.

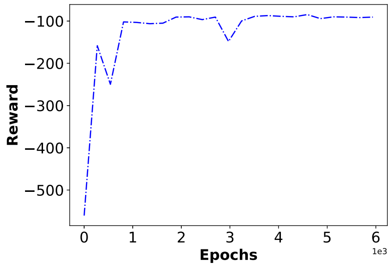


Fig. 10. Convergence of DRL in Section IV.

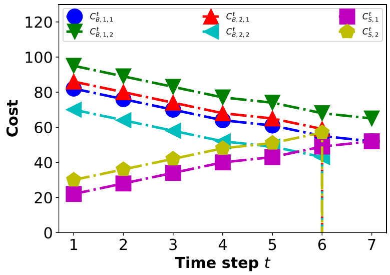


Fig. 11. Illustration of DRL double dutch auction result.

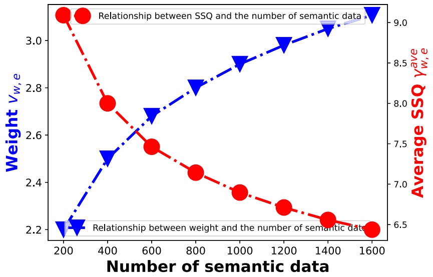


Fig. 12. Relationship between the weight $v_{w,e}$ and the number of semantic data required.

they are matched at $t = 7$ and the final trade price is determined from $t = 6$.

C. SSQ Relationship

We consider the set-up similar to Section VI-A to explain the relationship between SSQ, similarity score, and the number of semantic data required. We set $S_{1,1} = 0.8$ and vary the number of semantic data required. The relationship plot is shown in Fig. 12. When the number of semantic data increases, weight $v_{1,1}$ increases. An increase in $v_{1,1}$ refers to a decrease in SSQ score as the weight $v_{1,1}$ is inversely proportional to the SSQ score. However, when the number of semantic data required increases, the weight increases at a decreasing rate. This is due to the fact that insufficient data cause only a portion of the discomfort from the Metaverse application. A further increase in semantic data does not significantly decrease SSQ scores.

D. Monitoring the Tolerance Value θ_1 and θ_∞

In the next set-up, we monitor the relationship between the tolerance value θ_1, θ_∞, and the size of historical data. The

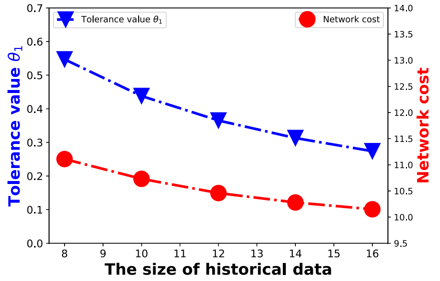


Fig. 13. Tolerance value θ_1 plot against the historical data size I'.

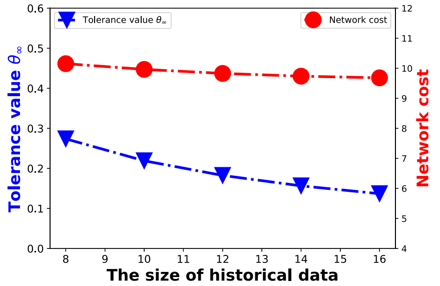


Fig. 14. Tolerance value θ_∞ plot against the historical data size I'.

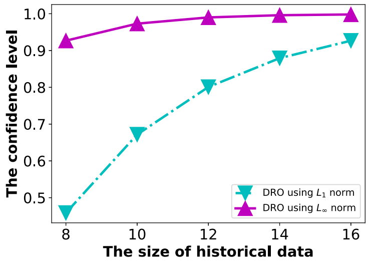


Fig. 15. Monitor the confidence level β_1 and β_∞.

confidence level is set at 0.95, $I = 2$, and the relationship curves are shown in Figs. 13 and 14. From the result, L_∞ norm metrics has a smaller tolerance value than L_1 norm and it is less conservative. This is because the confidence set will shrink as θ_∞ decreases which reduces the conservativeness. As historical data size grows, θ_1 and θ_∞ decrease, and more information is available, resulting in a tighter confidence set and a less conservative output solution and, therefore, a decrease in network cost. When I' increases to infinity, θ_1 and θ_∞ decrease to same value that is close to zero. The reference distribution can then be utilized as the actual true distribution after the confidence set becomes a singleton. Then, the VSPs can purchase lesser semantic data transmission with the reservation plan and purchase more with the on-demand plan only when the scenario occurs. It can be readily found that L_∞ can achieve a lower network cost.

E. Monitor the Confidence Level β_1 and β_∞

We monitor the confidence levels β_1 and β_∞ by fixing the tolerance values θ_1 and θ_∞ and varies the size of historical data I'. The confidence level is shown in Fig. 15. As the size of the historical data used increases, it is more confident that the constructed demand probability distribution contains the true distribution. DRO that uses L_∞ norm metrics can achieve a higher confidence level than DRO that uses β_1 norm metrics. Eventually, the confidence level β_1 and β_∞ will reach 1, but the L_∞ norm will converge faster.

TABLE III
DECISIONS OF VSPS [29]

Decision Variables	Values
$m^r_{1,2}$	1
$m^r_{2,3}$	1
$m^r_{3,1}$	1
$m^o_{1,1,3}(\lambda_1)$	97
$m^o_{1,2,1}(\lambda_1)$	234
$m^o_{3,1,2}(\lambda_2)$	316
$m^o_{1,2,3}(\lambda_2)$	4
$m^o_{1,3,1}(\lambda_2)$	27

F. Decision of VSPs

We consider a simple case using L_∞ norm with three VSPs, three edge sensing units, and two scenarios $|\lambda_i| = 2$ to show the decisions of the VSPs. There are three time slots representing 1: morning, 2: noon, and 3: afternoon. In λ_1, the VSPs require semantic data that are relevant to the vehicles driving on the road. While in λ_2, the VSPs require semantic data relevant to buses and traffic lights. The decisions of VSPs are shown in Table III. Each VSP subscribes to an edge sensing unit by using the reservation plan due to the low cost. Each edge sensing unit charges VSP with a different reservation cost, and the VSP subscribes to the edge sensing unit with the lowest cost. When the VSP has a demand and additional semantic data is required by using the on-demand plan, the VSP subscribes to the edge sensing unit according to the auction result. For example, in λ_1, VSP 1 is matched with edge sensing unit 1, VSP 2 is matched with edge sensing unit 3, and VSP 3 is matched with edge sensing unit 2. Both VSPs 1 and 2 request to transmit semantic data from time slot 1 as time slot 1 is morning, at the peak hour, and there are more vehicles on the road compared to other time slots.

G. Comparison With Other Schemes

We use a similar setting as in Section VI-D and set $n_e = 150$ and compare the DRO schemes (using L_1 and L_∞ norms) with deterministic optimization [24], random, and SIP [10] schemes. The deterministic optimization scheme uses the average SSQ scores, the number of semantic data required, and the similarity score $S_{w,e}$ to optimize the resource allocation. In the random scheme, the values of the decision variables are randomly generated. SIP uses historical data to perform the resource allocation, which can be solved by using (28)–(33). We use two types of scenarios $|\Omega\| = 2$. The VSPs have demand in both scenarios λ_1 and λ_2. However, the cost of the on-demand plan in λ_2 is higher than λ_1 as the users in λ_2 are faced with a lower QoE, i.e., the users in λ_1 have lower user demand than the users in λ_2. We compare the schemes with three types of SIP, i) uniform case, i.e., $P(\lambda_1) = 0.5$ and $P(\lambda_2) = 0.5$, ii) extreme case 1

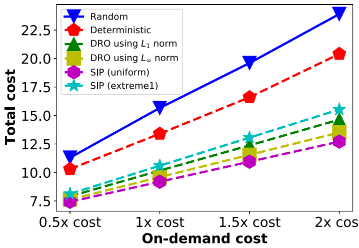


Fig. 16. Compare DRO with deterministic, random, SIP (uniform), SIP (extreme1) schemes.

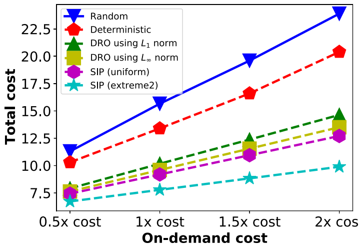


Fig. 17. Compare DRO with deterministic, random, SIP (uniform), SIP (extreme2) schemes.

TABLE IV
COMPARING SIP WITH DRO

Modeling Approach	**Difference in Accuracy**	**Computational Efficiency**
SIP (uniform)	-	0.341sec
DRO with L_∞	4.46%	36.44sec
DRO with L_1	10.66%	45.167sec

(high probability of higher user demand), i.e., $P(\lambda_1) = 0.3$ and $P(\lambda_2) = 0.7$, and iii) extreme case 2 (low probability of higher user demand), i.e., $P(\lambda_1) = 0.7$ and $P(\lambda_2) = 0.3$.

Fig. 16 compares the aforementioned schemes for extreme case 1 and Fig. 17 considers extreme case 2. As shown in the results, the deterministic and random schemes cannot adapt to the changes, so their costs are high. The SIP with a uniform case has a lower cost as compared to both DRO schemes. This is due to the fact that DRO uses the uniform distribution as a reference distribution to obtain the true distribution but the DRO is more conservative than SIP (uniform). This means that DRO offers higher system reliability (user satisfaction) at the expense of higher overall network costs. Table IV shows the comparison result between SIP (uniform) and DRO solutions. DRO with L_∞ has a tighter confidence set than that of L_1. Therefore, L_∞ has a lower difference in accuracy and computation efficiency as compared to L_1 norm. As an illustration, we also plot the SIP results (with extreme cases 1 and 2 known as the true distribution). For extreme case 1, the DRO with L_1 norm constructs a relatively close probability distribution of demand compared to DRO with L_∞ norm and SSP (uniform). The reason is that it is more conservative, therefore catering to extreme cases of higher user demand. However, for SIP (with extreme case 2 as the true distribution), we observe that the DRO results are further off due to conservativeness.

VII. Conclusion

In conclusion, we have proposed a two-phase stochastic semantic resource allocation (SSRA) scheme to allocate the semantic data captured by the edge sensing units. We have shown that energy consumption can be reduced by transmitting semantic data as compared to non-semantic data. Besides semantic communications, we further reduce the network's energy consumption by implementing the SSRA scheme. We have first formulated the SSRA scheme as a double dutch auction to match the VSP and the edge sensing unit. The matching is dynamic and depends on the quality of experience (QoE) from the Metaverse users and the semantic data transmission cost from the edge sensing unit. Then, we formulated DRO to minimize the expected operation cost (energy consumption) of the VSPs as not only is the demand of the VSPs uncertain, but the demand probability distribution is also unknown. Therefore, compared to the random and deterministic the SSRA scheme can achieve the best solution as it can adapt to changes in the probability distribution of VSPs' demands. If extreme case 1 is the actual distribution, then DRO with L_1 norm is better than L_∞ norm when constructing the probability distribution of demand. On the other hand, it is the other way around if extreme case 2 is the actual distribution.

References

[1] R. Mason, "Is the next generation of the Internet ready to be tested?," 2022. Accessed: Oct. 18, 2022. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2022/04/29/is-the-next-generation-of-the-internet-ready-to-be-tested/?sh=2db6cee5129b

[2] I. News, "AI: The driving force behind the metaverse?," 2022. Accessed: Oct. 18, 2022. [Online]. Available: https://www.itu.int/hub/2022/06/ai-driving-force-metaverse/

[3] C. Rachel, "The metaverse: Environmental costs of virtual reality," 2022. Accessed: Nov. 14, 2022. [Online]. Available: https://impakter.com/the-metaverse-environmental-costs-of-virtual-reality/

[4] P. Davies, "5G won't be enough: How are telecom companies building the metaverse with big tech?," 2022. Accessed: Oct. 18, 2022. [Online]. Available: https://www.euronews.com/next/2022/03/03/5g-won-t-be-enough-how-are-telecom-companies-building-the-metaverse-with-big-tech

[5] Y. Lin et al., "Meta-networking: Beyond the Shannon limit with multi-faceted information, semantic communication, IoV, multi-faceted inforamtion," Jun. 2023.

[6] M. Xu et al., "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Commun. Surveys Tut.*, vol. 25, no. 1, pp. 656–700, First Quarter, 2023.

[7] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.

[8] S. Dmytro Spilka, "How Big Data could form the cornerstone of the metaverse," 2022. Accessed: Aug. 14, 2022. [Online]. Available: https://venturebeat.com/datadecisionmakers/how-big-data-could-form-the-cornerstone-of-the-metaverse/

[9] D. Niyato, M. A. Alsheikh, P. Wang, D. I. Kim, and Z. Han, "Market model and optimal pricing scheme of Big Data and Internet of Things (IoT)," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[10] W. C. Ng, H. Du, W. Y. B. Lim, Z. Xiong, D. Niyato, and C. Miao, "Stochastic resource allocation for semantic communication-aided virtual transportation networks in the metaverse," 2022, *arXiv:2208.14661*.

[11] N. Stephenson, *Snow Crash: A Novel*. New York, NY, USA: Spectra, 2003.

[12] W. Y. B. Lim et al., "Realizing the metaverse with edge intelligence: A match made in heaven," 2022, *arXiv:2201.01634*.

[13] Y. Han, D. Niyato, C. Leung, C. Miao, and D. I. Kim, "A dynamic resource allocation framework for synchronizing metaverse with IoT service and data," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 1196–1201.

[14] W. C. Ng, W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, and C. Miao, "Unified resource allocation framework for the edge intelligence-enabled metaverse," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 5214–5219.

[15] C. Thompson, "South Korea launches metaverse replica of Seoul," 2023. Accessed: Sep. 01, 2023. [Online]. Available: https://www.coindesk.com/web3/2023/01/17/south-korea-launches-metaverse-replica-of-seoul/

[16] N. H. Chu et al., "MetaSlicing: A novel resource allocation framework for metaverse," *IEEE Trans. Mobile Comput.*, early access, Jun. 21, 2023, doi: 10.1109/TMC.2023.3288085.

[17] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, Apr. 2021, doi: 10.1109/TSP.2021.3071210.

[18] Y. Wang, M. Chen, W. Saad, T. Luo, S. Cui, and H. V. Poor, "Performance optimization for semantic communications: An attention-based learning approach," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv: 1810.04805*.

[21] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.

[22] H. Ren, C. Pan, K. Wang, W. Xu, M. Elkashlan, and A. Nallanathan, "Joint transmit power and placement optimization for URLLC-enabled UAV relay systems," *IEEE Trans. Veh. Technol*, vol. 69, no. 7, pp. 8003–8007, Jul. 2020.

[23] C. Zhao and Y. Guan, "Data-driven risk-averse stochastic optimization with Wasserstein metric," *Operations Res. Lett.*, vol. 46, no. 2, pp. 262–267, 2018.

[24] Y. Chen, B. Ai, Y. Niu, H. Zhang, and Z. Han, "Energy-constrained computation offloading in space-air-ground integrated networks using distributionally robust optimization," *IEEE Trans. Veh. Technol*, vol. 70, no. 11, pp. 12 113–12 125, Nov. 2021.

[25] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Proc. IEEE Asia-Pacific Serv. Comput. Conf.*, Singapore, 2009, pp. 103–110.

[26] R. Likamwa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, "Energy characterization and optimization of image sensing toward continuous mobile vision," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2013, pp. 69–82.

[27] Q.-V. Pham, M. Zeng, R. Ruby, T. Huynh-The, and W.-J. Hwang, "UAV communications for sustainable federated learning," 2021, *arXiv:2103.11073*.

[28] J. Won, D.-Y. Kim, Y.-I. Park, and J.-W. Lee, "A survey on UAV placement and trajectory optimization in communication networks: From the perspective of air-to-ground channel models," *ICT Exp.*, vol. 9, pp. 385–397, 2023.

[29] C. You and R. Zhang, "Hybrid offline-online design for UAV-enabled data harvesting in probabilistic LoS channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3753–3768, Jun. 2020.

[30] Y. Wu, Y. Wang, F. Zhou, and R. Q. Hu, "Computation efficiency maximization in OFDMA-based mobile edge computing networks," *IEEE Commun. Lett.*, vol. 24, no. 1, pp. 159–163, Jan. 2020.

[31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv: 1804.02767*.

[32] J. J. LaViola Jr, "A discussion of cybersickness in virtual environments," *ACM SIGCHI Bull.*, vol. 32, no. 1, pp. 47–56, 2000.

[33] S. Davis, K. Nesbitt, and E. Nalivaiko, "A systematic review of cybersickness," in *Proc. Conf. Interactive Entertainment*, 2014, pp. 1–9.

[34] E. Andrew and VitalBriefing, "NVIDIA researchers make real-time rendering a virtual reality," 2019. Accessed: Nov. 14, 2022. [Online]. Available: https://www.spiceworks.com/tech/innovation/articles/nvidia-researchers-make-real-time-rendering-a-virtual-reality/

[35] M. Shahid Anwar, J. Wang, S. Ahmad, A. Ullah, W. Khan, and Z. Fei, "Evaluating the factors affecting QoE of 360-degree videos and cybersickness levels predictions in virtual reality," *Electronics*, vol. 9, no. 9, 2020, Art. no. 1530.

[36] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *Int. J. Aviation Psychol.*, vol. 3, no. 3, pp. 203–220, 1993.

[37] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[38] W. Y. B. Lim et al., "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022.

[39] M. Xu, D. Niyato, J. Kang, Z. Xiong, C. Miao, and D. I. Kim, "Wireless edge-empowered metaverse: A learning-based incentive mechanism for virtual reality," 2021, *arXiv:2111.03776*.

[40] M. Gawinecki, P. Kobzdej, M. Ganzha, and M. Paprzycki, "Introducing interaction-based auctions into a model agent-based e-commerce system: Preliminary considerations," in *Proc. Euro Amer. Conf. Telematics Inf. Syst.*, 2007, pp. 1–8.

[41] V. Jain and B. Kumar, "Auction based cost-efficient resource allocation by utilizing blockchain in fog computing," *Trans. Emerg. Telecommun. Technol.*, vol. 33, 2022, Art. no. e4469.

[42] H. Li, H. Gao, T. Lv, and Y. Lu, "Deep Q-learning based dynamic resource allocation for self-powered ultra-dense networks," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2018, pp. 1–6.

[43] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3177–3192, Oct. 2021.

[44] W. C. Ng et al., "Stochastic coded offloading scheme for unmanned-aerial-vehicle-assisted edge computing," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5626–5643, Apr. 2023.

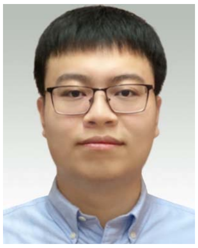
[45] Grover, "Smartphone rental," 2023. Accessed: Jul. 10, 2022. [Online]. Available: https://www.grover.com/us-en/phones-and-tablets/smartphones

[46] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-YOLO: Spiking neural network for energy-efficient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11 270–11 277.

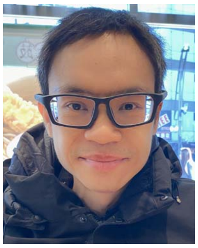
[47] Y. Mao, "A memory-efficient YOLO object detection convolutional neural network inference engine on the KiloCore 2 manycore platform," PhD dissertation, 2021. [Online]. Available: https://escholarship.org/uc/item/1b5393kk

Wei Chong Ng received the BEng degree in electrical and electronic engineering (highest distinction) from Nanyang Technological University, Singapore, in 2020. He is currently working toward the PhD degree with Alibaba Group and Alibaba-NTU Joint Research Institute, Nanyang Technological University, Singapore. His research interests include the Metaverse, stochastic integer programming, and edge computing.

Wei Yang Bryan Lim received the PhD degree from the Nanyang Technological University (NTU), Singapore, in 2022 under the Alibaba PhD Talent Programme and was affiliated with the CityBrain team of DAMO academy during this period. He is an assistant professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, having joined in 2023. Between 2022 and 2023, he held the title of Wallenberg-NTU presidential postdoctoral fellow. His doctoral efforts earned him accolades, such as the "Most Promising Industrial Postgraduate Programme Student" award and the IEEE Technical Community on Scalable Computing (TCSC) Outstanding PhD Dissertation Award. Besides, his works have been honored with Best Paper Awards, notably from the IEEE Wireless Communications and Networking Conference (WCNC) and the IEEE Asia Pacific Board. He also serves on the Technical Programme Committee for flagship conferences and is a review board member or guest editor for reputable journals like *IEEE Transactions on Parallel and Distributed Systems* and *IEEE Wireless Communications*. In 2023, he co-edited a submission on "Requirements and Design Criteria for Sustainable Metaverse Systems" for the International Telecommunication Union (ITU). Alongside his publications in high impact journals of his field, he is the first author of two books. Additionally, he has been a visiting scholar with the KTH Royal Institute of Technology, University of Sydney, University of New South Wales, and the Singapore University of Technology and Design.



Zehui Xiong received the PhD degree from Nanyang Technological University, Singapore. He is currently an assistant professor with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design. Prior to that, he was a researcher with Alibaba-NTU Joint Research Institute, Singapore. He was the visiting scholar with Princeton University and University of Waterloo. His research interests include wireless communications, network games and economics, blockchain, and edge intelligence. He has published more than 140 research papers in leading journals and flagship conferences and many of them are ESI Highly Cited Papers. He has won more than 10 Best Paper Awards in international conferences and is listed in the World's Top 2% Scientists identified by Stanford University. He is now serving as the editor or guest editor for many leading journals including the *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Vehicular Technology*, *IEEE Internet of Things Journal*, *IEEE Transactions on Cognitive Communications and Networking*, *IEEE Transactions on Network Science and Engineering*, *IEEE Systems Journal*. He is the recipient of IEEE TCSC Early Career Researcher Award for Excellence in Scalable Computing, IEEE CSIM Technical Committee Best Journal Paper Award, IEEE SPCC Technical Committee Best Paper Award, IEEE VTS Singapore Best Paper Award, Chinese Government Award for Outstanding Students Abroad, and NTU SCSE Best PhD Thesis Runner-Up Award. He is the founding vice chair of Special Interest Group on Wireless Blockchain Networks in IEEE Cognitive Networks Technical Committee.



Dusit Niyato (Fellow, IEEE) received the BEng degree from the King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand, in 1999, and the PhD degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include the areas of Internet of Things (IoT), machine learning, and incentive mechanism design.

Xuemin Sherman Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks. He is a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada fellow, a Canadian Academy of Engineering fellow, a Royal Society of Canada fellow, a Chinese Academy of Engineering foreign member, and a distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee chair/co-chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the president of the IEEE ComSoc. He was the vice president for Technical $\&amp;$ Educational Activities, vice president for Publications, Member-at-Large on the Board of Governors, chair of the Distinguished Lecturer Selection Committee, and member of IEEE Fellow Selection Committee of the ComSoc. He served as the editor-in-chief of the *IEEE Internet of Things Journal*, *IEEE Network*, and *IET Communications*.

Chunyan Miao (Senior Member, IEEE) received the BS degree from Shandong University, Jinan, China, in 1988, and the MS and PhD degrees from Nanyang Technological University, Singapore, in 1998 and 2003, respectively. She is currently a professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU), and the director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY). Her research focus on infusing intelligent agents into interactive new media (virtual, mixed, mobile, and pervasive media) to create novel experiences and dimensions in game design, interactive narrative, and other real world agent systems.