



Energy-efficient secure dynamic service migration for edge-based 3-D networks

Guhan Zheng¹ · Keivan Navaie¹ · Qiang Ni¹ · Haris Pervaiz² · Charilaos Zarakovitis³

Accepted: 20 December 2023 / Published online: 28 January 2024
© The Author(s) 2024

Abstract

In communication networks, where users are highly mobile, migrating edge servers that are performing services closer to users, i.e., service migration, is essential to maintain the high quality of service (QoS). However, existing dynamic service migration techniques face two distinct challenges: (1) The security and energy consumption of service migration systems need to be optimised urgently; (2) The uncertainty of user movement makes it difficult to develop optimal service migration strategies, especially in future three-dimensional (3-D) communication networks. To address these challenges, we propose a novel energy-efficient secure 3-D dynamic service migration framework for communication networks. We then quantify the cost of service migration based on the proposed framework considering security, energy efficiency and delay and present a solution based on a deep reinforcement learning (DRL) approach to make migration decisions optimally in the 3-D communication network. We also propose a universal formula for measuring the reliability value of intelligent autonomous nodes in order to reduce the energy consumption and delay of the proposed security paradigm and to optimise the service migration decision making. Simulation results demonstrate our proposed migration strategy for 3-D communication network services outperforms the baseline solutions in terms of reducing communication network delay and energy consumption while preserving migration security. Moreover, the results confirm the effectiveness of the proposed reliability value calculation approach applied to improve the QoS in the secured edge networks.

Keywords Service migration · Blockchain · Reliability value · 3-D communication networks · Deep reinforcement learning (DRL)

1 Introduction

With the continuous development of mobile wireless communications and the Internet-of-Things (IoT), cities are becoming smarter, for example, autonomous vehicles with high mobility are becoming a reality. The number of connected and/or autonomous vehicles on the road is anticipated to increase substantially in the coming years [1]. These vehicles and users have versatile resource requirements in terms of computing, communication and storage and in most cases are extremely latency-sensitive. These create new challenges on resource-constrained communication networks [2]. Mobile edge computing (MEC) [3, 4] is considered a promising approach that integrates edge servers into communication network facilities such as base stations (BSs) or roadside units (RSUs). This enables users with quick access to computation and data storage resources. Moreover, edge servers can also be integrated into the unmanned aerial vehicle (UAV) base stations [5] to further improve flexibility and radio cov-

✉ Qiang Ni
q.ni@lancaster.ac.uk
Guhan Zheng
g.zheng2@lancaster.ac.uk
Keivan Navaie
k.navaie@lancaster.ac.uk
Haris Pervaiz
haris.pervaiz@essex.ac.uk
Charilaos Zarakovitis
c.zarakovitis@iit.demokritos.gr

¹ School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, UK

² School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK

³ National Centre for Scientific Research Demokritos, Athens, Greece

erage for supporting remote areas hence enabling ubiquitous service provisioning.

The deployment of MEC brings cloud services closer to the users and significantly reduces service access latency [6]. Nevertheless, serving highly mobile users such as vehicles requires careful planning and updating the placement of edge services, because such users are likely to be moving away from the associated edge servers due to their mobility. Hence, without timely migration of provision quality might be degraded resulting e.g., not corresponding edge services, and slower response times for latency-sensitive applications [7].

To address this issue, the service needs to migrate to a new service location to enable the user to maintain the advantages of MEC and the QoS of future communication networks (Fig. 1). In contrast to the cellular handover, the users in service migration can continue receiving service from original edge servers when no longer connected to the original base stations. Because a remote edge server can communicate with users through currently associated base stations and backhaul networks, the edge service can be migrated to any potential edge server. Cellular handover is often decided by signal strengths and must occur when the original base station signal does not satisfy users. It has a smaller decision space than service migration.

Service migration decreases the transmission latency between the user and the edge server hosting its service. Nevertheless, it can also incur extra service interruption, energy consumption and network overhead. Therefore, further investigations are required to evaluate when and where to migrate the service to achieve a balance between “migration cost” and “transmission cost”. In addition, the migration process may cause unprecedented vulnerabilities and flaws in the MEC environment, e.g., malicious code injection attacks, and violations of privacy in terms of location and usage patterns, making it one of the weakest phases of MEC [8]. Secure service migration thus becomes an important requirement for vehicle edge service continuity, hence the trade-off between security, energy consumption and latency of service migration should be further investigated.

In recent years, various research efforts have been made on the development of dynamic service migration strategies for mobile users. For instance, one-dimensional (1-D) environment migration strategies were first proposed in [9]. They constructed the migration decision processes as Markov Decision Processes (MDPs) that enable predict users’ next move. They thus developed corresponding migration strategies and addressed users’ mobility challenges during service migration. However, in these approaches, migration strategies are based on 1-D MDPs and clients are considered to only move straight ahead. Two-dimensional (2-D) mobility MDP migration strategy was then presented in [10]. It approximates the underlying state space via the distance

between the user and the location of the service. User’s movement in a 2-D space thus can be anticipated. However, it acts as a 1-D service migration strategy essentially, where the error increases as the operational distance of the vehicle increases and is also unable to change strategy in real time according to network conditions. In future communication systems, the networks that employ UAVs become more sophisticated dynamic three-dimensional (3-D) networks. Hence, the development of a more accurate, secure and energy-efficient 3-D environmental dynamic migration strategy is more necessary to enable real-time service provisioning in such a fast changing environment. Further, to the best of our knowledge, few investigations have considered a specific security measure and its cost in 3-D network dynamic service migration.

To address these urgent needs, a novel blockchain technology based energy-efficient secure service migration framework is proposed to ensure the security of 3-D service migration. Based on the proposed framework, a general reliability value calculation scheme for intelligent autonomous nodes is designed to decrease the energy consumption of the presented framework. Furthermore, a new 3-D communication network service migration strategy is proposed that optimally the security cost and energy consumption of a service migration system while guaranteeing a low migration delay. The corresponding dynamic optimisation problem is then expressed as an MDP with the same general reliability value formula as we proposed for intelligent autonomous nodes to enable quantifying of the migration cost. Finally, an efficient deep reinforcement learning (DRL)-based scheduling algorithm is developed to solve this optimisation problem.

The main contributions of this paper are summarised as follows.

(1) We propose a novel energy-efficient secure service migration framework for 3-D communication networks based on blockchain that improves the security, reliability and energy efficiency of service migration between the MEC servers.

(2) We present a universal intelligent autonomous nodes reliability assessment approach that considers nodes’ security and efficiency. We further apply this approach to the suggested framework and service migration selection strategy to improve network efficiency and reduce energy consumption.

(3) A 3-D dynamic service migration strategy is also proposed which considers UAV base stations. This strategy is modelled as an MDP in which a service migration reward function considers security, energy consumption and quality-of-service (QoS) when migrating services. The optimal service migration strategy is obtained using the DRL approach.

The rest of the paper is organized as follows. Section 2 summarizes the existing research findings related to this work. We then introduce the security system model in Sect. 3

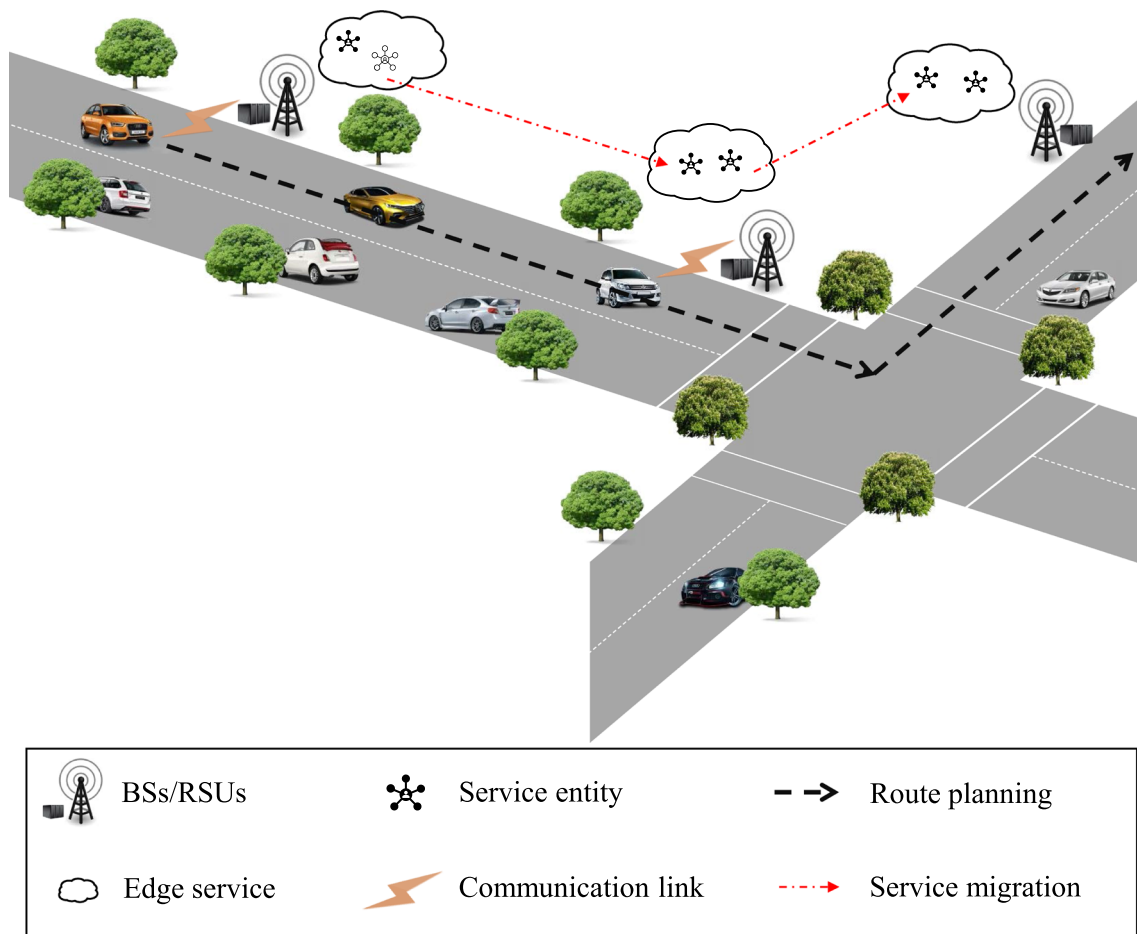


Fig. 1 An example of service migration application scenario

and the reliable value formula in Sect. 4. In Sect. 5, the migration strategy for 3-D vehicle networks is introduced. The performance of the proposed mechanism and method are then evaluated and analyzed by simulations in Sect. 6. Finally, we conclude the paper in Sect. 7.

For easy reference, the main parameters used throughout this paper are presented in Table 1.

2 Related works

2.1 Service migration strategies

Several existing works are focused on latency reduction for making implementing service migration decisions. An early effective migration mechanism was proposed in [11], which implies the service always follows with the user, i.e., “always migration”. In [12], a short-sighted strategy to determine whether to migrate the service based on the current service condition of the user called “myopic” was presented. None of the above works however consider users’ mobility predic-

tion. As mentioned before, the authors in [9] first proposed a 1-D service migration moving model based on 1-D MDP and demonstrated the presence of an optimal threshold policy to optimize the total cost of service migration. However, these strategies could not be adapted to more complex multidimensional scenarios. S. Wang et al. in [10] extended the application scenario to 2-D mobility space using the distance-based MDP they presented to develop a more general and efficient service migration strategy. They further described a general cost model based on distance and introduced a mathematical framework to achieve the best migration strategy. Moreover, they also examined the application of the proposed 2-D migration algorithm using the real date of the San Francisco taxi movement. The error however increases with environmental changes and the distance travelled by taxi since distance-based MDP is 1-D MDP essentially. Similar 2-D service migration was further investigated in [13] and [14].

The migration energy consumption and latency balance have been also investigated. For instance, [15] proposed a migration energy consumption model. Further, this model

Table 1 Notation definition

Symbol	Definition
L	The set of all-possible user locations
μ_t	The user location at time t
h_t	The service location at time t
d_t	The distance between user and service (hops)
s_t	The real-time system state
x	Service migration distance (hops)
y	Transmission distance (hops)
k	Number of blockchain nodes
V	Number of communications in blockchain system
p	View change probability
T_r	Reliability value
T_o	Operating speed value
T_s	Available resources and security value
T_h	Hardware performance value
T_c	Algorithm performance value
O_v	Algorithm objective value
S_v	Algorithm subjective value
Ω	Complexity indicator
$M(x)$	Migration cost (s)
$T(y)$	Transmission cost (s)
$B(x)$	Security cost (s)
μ_t	Available bandwidth of target server
N	Application-specific maximum allowed distance
n	Tier, the states with the same value
$O(N)$	The total number of nodes in range N
$O(n)$	The number of nodes in the n
r	The probability of user movement
Ψ	The distance between two states

however favours the transfer of uncompleted tasks of users rather than service migration. [16] took a similar energy consumption model as [15] and used deep Q network (DQN) to estimate the optimal migration policy. Artificial intelligence (AI) techniques such as DQN have been used in recent years to enhance the ability of migration optimization models. A DQN method is presented in [17] that lower latency service migration strategies can be obtained compared to dynamic programming. The service can only choose to migrate to the client's current location server rather than the most suitable server that may exist in the path. The strategies in this method are sub-optimal. Similarly, [18] also designed a reward function and used DQN to search for the optimal migration strategy. Nevertheless, none of the above works using the DQN approach mentions user movement patterns and the impact of adopting such a machine learning technique on the node.

2.2 Service migration security

Research on the security of service migration is much less investigated than that of optimising migration strategy. For instance [19] brought the cost of security into the service migration decision making in terms of distance, but only an approximate model of the safety costs was incorporated. Similarly, [20] introduced a platform of trust credentials, which does not require an active third party, and ensures service migration to a trusted cloud platform. However, no results were presented to support their framework to ensure their platform is secure enough. [21] suggested blockchain could be used to tackle the trust issue in the migration of entities in different domains but did not provide specific measures for implementation. In addition, none of the introduced security frameworks take into account the energy consumption associated with security measures.

3 System model

In this section, we present the proposed migration framework based on the blockchain system in detail.

3.1 Migration model

We consider a user moving in a 2-D geographic zone accessing randomly distributed fixed/UAV-assisted edge cloud servers. These servers are distributed in a 3-D space. All possible locations of the user consist of set L , where L is a finite set that can be arbitrarily large. In each time slot, the user location $l \in L$ can be represented as a 2-D vector as it moves in a 2-D space, i.e., (a, b) . In addition, we assume each user's location has an edge facility that can provide services in the cellular network, and the user's location and the 2-D vectors (i.e., height is 0) of the edge facility are overlapping. Thus, $\|l_1 - l_2\|$ is as the distance between l_1 and l_2 . Note that the distance metric is not the Euclidean distance but the number of hops from one edge facility to another. To facilitate differentiation, we denote the user's location at the time slot t as $\mu_t = (a_\mu, b_\mu)|c_\mu$, where c_μ is the height of users, and $c_\mu = 0$. Similarly, the location of the current service facility at the time slot t can be represented as $h_t = (a_h, b_h)|c_h$. Therefore, we use $s_t = \{\mu_t, h_t\}$ to indicate the real-time status of the system and the next time status is $s_{t+1} = \{\mu_{t+1}, h_{t+1}\}$.

According to the Markov mobility model, the users' positions remain fixed for the duration of a time slot and only change from one time slot to the next. The time slot model is similar to the sampling in continuous time, where the duration of the time slot can be variable. At the beginning of each time slot, the edge facility has the following two options:

1) Migrate service from h_t to other appropriate location h'_t , which will result in a migration cost M and security cost B (details are elaborated in the subsequent section). Migration cost M is defined as non-zero service interruption time similar to [22, 23]. It is related to the size of the server's random access memory (RAM) and the available bandwidth [24, 25]. As each migration entails a certain cost, we assume that it is also related to the number of migrations. We use $M(x)$ to represent total migration cost, where $x = \|h_t - h'_t\|$ is the distance between h_t and h'_t . After completing the migration, the system state changes as $s'_t = \{\mu_t, h'_t\}$, and the next time slot service location is $h_{t+1} = h'_t$.

2) No service migration. State maintained at $s_t = \{\mu_t, h_t\}$ and the next time slot state s_{t+1} will be changed as $\{\mu_{t+1}, h_{t+1} = h_t\}$, hence no migration cost.

However, if a user is not at the location of the current service facility, there also exists a data transmission cost of T after possible migration. In general, in a 2-D geographical space, transmission cost is distance-dependent and can capture the delay of data transmission such as transmission delay, propagation delay and queuing delay. It can be defined as $T(y)$, where y is the distance between the user location and the current service location. In traditional 2D service migration, y is calculated as the number of hops between the user location and the current service location, i.e., $\|h'_t - \mu_t\|$. Furthermore, queuing delay represents the time a user requests in the request queue to queue in line for a free resource. It is sensible to assume that more users in the same base station range may have more service requests resulting in higher queuing delays and transmission costs. Since UAV-assisted base station has wider coverage and may serve more clients at the same time. Therefore, the distance y in 3-D scenarios can be rewritten as:

$$y = d + \sum_{i=1}^e \alpha c_i, \quad (1)$$

where d and e are the numbers of fixed base stations and UAV-assisted base stations in the route respectively. We have $d + e = \|h'_t - \mu_t\|$ in 3-D scenarios. Further, in Eq. (1), α is the weighting parameter and c_i is the height of UAVs. As UAV-based base stations provide a larger coverage area hence requiring longer propagation distances and serving more users. These characteristics are merely due to the addition of the UAV altitude parameter. The change in altitude can thus affect the propagation delay and the transmission delay. Moreover, it is shown that for service migration, the major factors affecting the migration delay are the transmission and propagation delays. Therefore, we only consider the height parameters in this paper to fit the 3-D service migration scenario and to reduce the modelling complexity.

Moreover, we assume that the time spent due to transmission and migration costs is much less than that of the length

of each time slot. Therefore, the costs do not vary with the duration of the time slot.

3.2 Blockchain system

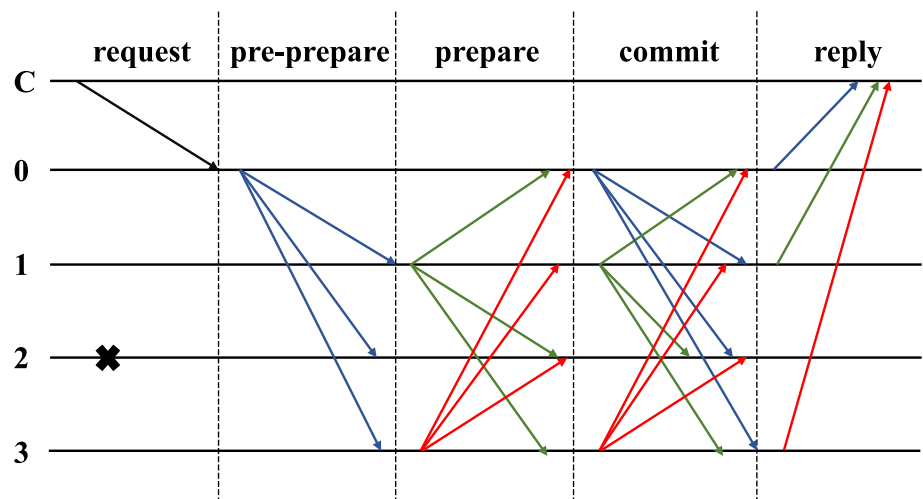
Blockchain provides significant protection of customer privacy and security. However, commonly used proof based blockchain consensus mechanisms such as Proof-of-Work (PoW) and Proof-of-Stake (PoS) have high energy consumption, long transaction confirmation latency and low throughput [26]. Thus, in our energy-efficient secure migration framework, we employ the Practical Byzantine fault tolerance (PBFT) consensus mechanism that is well suited for IoT as it offers low computational power and complexity [27, 28].

In the proposed framework, all blockchain nodes consist of edge facilities. The total number of nodes is k . These nodes are divided into two types: a primary node and a backup node. The primary node is responsible for assigning serial numbers to the client's transaction requests and forwarding them to the backup node. The backup nodes are responsible for performing the requests according to the serial numbers assigned by the primary node, checking the legitimacy of the serial numbers, and using a timeout mechanism to monitor whether the primary node is an anomaly. In addition, the maximum number of fault-tolerant nodes is defined as f , where $3f + 1 < k$ [29].

The main consensus process of the PBFT algorithm is divided into 3 phases (Fig. 2): *pre-prepare*, *prepare*, and *commit* [29]. During the *pre-prepare* phase, after receiving a request from the client, the primary node assigns a serial number to the request and broadcasts a "pre-prepare" message to all backup nodes. As soon as a backup node recognizes the serial number assigned by the primary node and accepts the "pre-prepare" message, this backup node enters the *pre-prepare* phase. In the *prepare* phase, the backup node sends a "prepare" message to all nodes and each node receives "prepare" messages from the other nodes. Upon receiving $2f$ "prepare" messages (including itself) that match the "pre-prepare" message, the backup node enters the third phase and the request is considered *prepared*. Meanwhile, the backup node broadcasts a network-wide "commit" message and the request moves to the *committed* state and is processed upon receiving no less than $k - f$ authenticated messages (including itself). The result of the operation is returned to the client and stored in the local status database.

If the primary node is in an abnormal condition, or running slow, the primary node is changed. We name this primary node change process as "view change" and a primary node effective time is a "view". In view change progress, all backup nodes broadcast the view change message and the number of communications is $(k - 1)^2$. When a "view change" is confirmed, the primary node announces a view

Fig. 2 PBFT algorithm operating under normal conditions [29]



change confirmation message to all backup nodes, which passes $k - 1$ messages. We assume each node has the same communication condition. The transmission rate of a single communication therefore can be denoted by

$$r = B_0 \log_2 \left(1 + \frac{p_0 h_0}{\sigma_0} \right), \quad (2)$$

where B_0 is the bandwidth, p_0 is the transmission power, h_0 is the channel gain and σ_0 is the channel noise. Therefore we can consider that the number of communications in the network is proportional to the energy consumption of the network.

Normally, in the original PBFT, other nodes elect a new leader based on the sequence of the nodes, following:

$$\text{primaryId} = v \bmod k, \quad (3)$$

where v is the reference number of views, primaryId is the reference number of nodes and mod is modular arithmetic. In this setting, each node has an equivalent probability of acting as a primary node. Nevertheless, it does not take into account the real operational capacity and security of the node. This increases the likelihood of view changes. Hence introducing additional communication consumption and latency.

4 Reliability value calculation in blockchain-enabled service migration systems

4.1 Reliability value calculation

For secure and energy-efficient communication, the blockchain node with a high level of processing capacity and security should be chosen as the primary node. Many similar scenarios also exist in future IoT networks aimed at finding more

reliable nodes. These reliability nodes usually have better operating speed, available resources and security. We, therefore, propose a universal reliability node evaluation method jointly considering node operating speed, available resources and security for future IoT networks. The evaluation value of each node is defined as the node's reliability value. The reliability value T_r can be thus expressed as:

$$T_r = T_o T_s, \quad (4)$$

where $T_o > 0$ and $T_s > 0$ are the operating speed value, and available resources and security value of the node respectively.

Node operating speed is normally determined by both the node hardware and the node algorithms used, we have

$$T_o = \beta T_h + (1 - \beta) T_c, \quad (5)$$

where β is the weight parameter. $T_h > 0$ is the node hardware performance value that depends on the processing unit's cycles. Since the values are used for numerical evaluation without a unit, we only select numerical values when the unit is in Gcycles/s, which is generally less than 10. $T_c > 0$ is the value related to algorithms used by the node. As the whole node algorithms are numerous and mostly not independent of each other. This is however difficult and unrealistic to consider them all for a universal node evaluation. Fortunately, the ubiquitous AI will assist and enable IoT nodes as intelligent autonomous nodes in the future IoT network. Furthermore, AI and machine learning techniques are anticipated to be one of the main algorithms applied in intelligent autonomous nodes. Machine learning techniques, especially deep learning techniques represent to a great extent the processing capability of intelligent autonomous nodes. We hence denote T_c as the value associated with the machine learning algorithms used by the node.

Generally, a good machine learning algorithm for humans should consider the following domains [30]: accuracy, appropriate data and algorithm, robustness, privacy protection, context awareness, subjectivity and computational overhead. Obviously, some domains such as accuracy and robustness can be judged by running and understanding the algorithm itself. Other domains such as context-awareness and subjectivity are up to the users to judge after the experience. Therefore, we define the T_c calculation formula as:

$$T_c = \gamma O_v + (1 - \gamma)S_v, \tag{6}$$

where O_v is the algorithm objective value and S_v is the subjective value judged by humans, which is adjusted by coefficient γ . Here we can define O_v as:

$$O_v = \frac{1}{N} \sum_{m_n \in m_N} \frac{a_{m_n} r_{m_n} d_{m_n}}{\Omega_{g_n} \omega_{m_n}}, \tag{7}$$

where $m_n \in \{m_1, m_2, \dots, m_N\}$ represents the evaluated machine learning model in an intelligent autonomous node. a_{m_n} is a accuracy indicator in $[0, 1]$, which means the model m_n 's accuracy after training with appropriate labels. $r_{m_n} \in [0, 1]$ denote the node robustness, indicating the accuracy after the algorithm withstanding a degree of attack. $d_{m_n} \in [0, 1]$ measures the appropriateness of the training data, the larger number implies the more perfect the data set. It can be quantified based on the real-time training label situation, such as data balance situation and independent and identically distributed situation.

In addition, we adopt the model complexity[31] as the denominator to denote the computational overhead of the node. Complexity indicates, to some extent, how fast an algorithm can be operated. Different from run time, complexity is not judged in relation to hardware and is more aligned with our evaluation of the algorithm itself. Furthermore, the provenance of reasoning (e.g., which data features caused which decisions) is also one of the key concerns for artificial intelligence implementation. Simpler models enable better emotional judgement of nodes by humans. In the formula, g_n is the model type of m_n and ω_{m_n} quantifies the structural complexity for the model (such as the number of neurons in deep learning, and depth in decision trees). Ω is a complexity indicator.

The subjective value parameter S_v is determined by the sense from human perception of the advantages and disadvantages of intelligence in a 6 G environment. We can express this by

$$S_v = \frac{1}{\Gamma N} \sum_{\kappa_\Gamma \in \Gamma} \sum_{m_n \in m_N} p_{\kappa_\Gamma m_n} e_{\kappa_\Gamma m_n} s_{\kappa_\Gamma m_n}, \tag{8}$$

where Γ is the number of participants, $\kappa_\Gamma = \{\kappa_1, \kappa_2 \dots \kappa_\Gamma\}$, $p_{\kappa_\Gamma m_n}$, $e_{\kappa_\Gamma m_n}$ and $s_{\kappa_\Gamma m_n}$ represent the privacy-preserving ability, the context-awareness, and the overall subjective perception of model m_n as judged by user κ , respectively.

In addition, to determine the real-time security, efficiency and available computing resources of the node, we utilize node energy level as available resources and security value to help determine if there is a problem with a node or not. In cases where nodes are maliciously intruded or malicious nodes launch a malicious attack, they normally consume a large amount of energy. When a node has a high workload to cope with, the energy consumption ratio is also greater. We define the residual energy E_{res} and total energy level E_{tot} . The value T_s is given by

$$T_s = \begin{cases} \frac{E_{res}}{E_{tot}}, & \text{if } \frac{E_{res}}{E_{tot}} > \zeta, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where ζ is the security energy threshold. The node is insecure upon falling below the threshold. The node is also considered to be unreliable in this case. We can therefore obtain the approach to evaluate the reliability node, which is denoted as

$$T_r = \begin{cases} (\beta T_h + (1 - \beta)T_c)T_s, & \text{if } T_s > \zeta, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

4.2 Complexity analysis

For the reliability value evaluation complexity, since it does not vary with variables, we can have the complexity as $O(1)$. Therefore, in this subsection, we discuss the number of basic executions while integrating the proposed reliability value in PBFT.

Based on Sect. 3.2, we have the total number of passed messages after completing the three-phase consensus process, z , is

$$z = 2k^2 - 2k. \tag{11}$$

In addition, We assume the probability of view change in a time slot is p . The total number of communications V for a given probability p of view changes thus is

$$V = z + pk(k - 1) = 2k^2 - 2k + pk(k - 1). \tag{12}$$

In our proposed framework, a new leader selection is based on ‘‘reliability value’’. All backup nodes should broadcast the view change message and the reliability values of neighbouring nodes during the view change. It is used to elect the node with higher operational speed and better security to reduce p , so as to avoid potential threats and extra energy consumption. Therefore, the addition of the reliability value increases

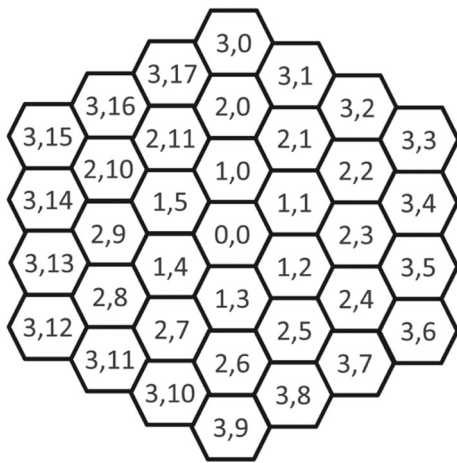


Fig. 3 Example of a 3-tier 2-D model on hexagon cells [10]

the number of calculations k but saves communication consumption of $pk(k-1)$.

5 Optimal policy for offset-based MDP

In the following section, we elaborate on how to develop an optimal service migration strategy for mobile users in a 3-D environment considering security, energy and QoS.

5.1 Control decisions and costs

As previously stated (Sect. III-A), in each time slot, servers have the following two choices: service migration or no migration. Whichever choice is made, it may bring a certain amount of cost. We define the migration cost, $M(x)$, and the transmission cost, $T(y)$, according to migration distance x and transmission distance y , as the following:

$$M(x) = \begin{cases} 0, & \text{if } x = 0, \\ (\frac{1}{\mu_t} + \delta)bx, & \text{if } x > 0, \end{cases} \quad (13)$$

$$T(y) \triangleq \begin{cases} 0, & \text{if } y = 0, \\ \rho_c + \rho_l \theta^y, & \text{if } y > 0, \end{cases} \quad (14)$$

where $T(y)$ is a general non-decreasing function defined in the form of constant plus exponent to match to transmission cost. This comparable transmission cost is widely used in service migration decision making [16, 18]. Thus ρ_c , ρ_l , and θ are real-valued parameters. Further, δ is also a real-valued parameter related to the server's RAM. μ_t is the available bandwidth of the target server and b is the conversion parameter indicating the relationship between bandwidth and latency. That is because the cost of migration is dependent on the size of the server's RAM and its available bandwidth. Changes in the RAM size have little effect on latency and energy consumption, so it is generally set as a constant value

[32]. Differently, as the available bandwidth increases, the migration time and energy consumption decrease exponentially and eventually approach a constant value [24, 25]. For the sake of simplicity, hereafter we assume that the relationship between migration cost and the available bandwidth follows $1/\mu_t$.

However, since blockchain was employed to secure services during migration, the security costs resulting from the blockchain cannot be ignored, which is

$$B(x) = \vartheta x, \quad (15)$$

where ϑ is a constant with the unit in second, which represents the migration delays caused by blockchain in each service migration. That is because it is shown in blockchain systems based on PBFT [33] that the latency for per-block generation remains almost constant at around one second for a limited transaction capacity. Also, the transaction capacity is much larger than the number of transactions per second in our range of service migration. Thus, the overall cost (unit in seconds) for a one-time slot can be denoted by

$$C = M(x) + B(x) + T(y), \quad (16)$$

5.2 Off-set based MDP

As clients move in a 2-D geographical area, we consider a hexagonal cell structure, where the user fits a uniform 2-D random walk Markov mobility model in an infinite space to approximate real-world movement trajectory. We define the tier $N = \{1, 2, \dots, n\}$ as an application-specific maximum allowed distance and d_t as the number of hops between the user and the service before possible migration. We always perform migration when $d_t > N$, so that we only focus on states $d_t < N$. The relation between N and the total number of nodes is

$$O(N) = \frac{6N(N+1)}{2} + 1. \quad (17)$$

The number of nodes in the tier n is

$$O(n) = 6n. \quad (18)$$

Assume that moving to one of six neighbours with probability r and staying in the same cell with probability $1-6r$. In all previous research, a user was only considered to have three movement options based on approximation by distance-based MDP when $tier > 0$. Stay at the same tier, move to a lower tier, and move to a higher tier with the probability of $1-4r$, $1.5r$, and $2.5r$ separately. Because the probability of moving to a lower tier is either r or $2r$, while to a higher tier is either $2r$ or $3r$, averaging the probabilities yields. Obviously, the results become increasingly inaccurate as the movement

time increases. It's not suitable for more complex 3-D communication networks.

We assume the client has seven options for movement. If the client is not at (0,0), there are two different location types of nodes. One type includes the node such as (1,1), which has a probability movement of r moving to a lower tier and a probability movement of $3r$ moving to a higher tier. The other type is the node like (2,1), which has a probability movement of $2r$ moving to a lower tier and a probability movement of $2r$ moving to a higher tier. We thus introduce polar coordinates to determine the location type of the node. As mentioned earlier, the 2-D vector of each node can be represented as (a, b) . Here, a is the number of tiers at the location of the node, and b is the location number within each tier. Assuming the node coordinates (a, b) are labelled as in Fig. 3. Since nodes in $n=1$ are all the first type, when $n > 2$, if the result of $\frac{2\pi j}{2\pi - 1}$ is an integer, then the node is the first type or the second. And state in $n + 1$ or $n - 1$ can also be identified by the answer of $\frac{2\pi j}{\frac{n}{n-1}}$. In addition, the distance between two states can be denoted by

$$\psi = \sqrt{n_1^2 - 2n_1n_2\cos(\varphi_1 - \varphi_2) + n_2^2}, \quad (19)$$

where φ is the angle in the polar coordinate, obtained from state j times $\frac{2\pi j}{n}$.

5.3 Optimal solution

We model the service migration policy as MDP. As the state in one time slot t is $s_t = \{u_t, h_t\}$ that $s_{(t+1)} = \{u_{(t+1)}, h_{(t+1)}\}$ is for the next time slot $t+1$. After the action a choose the newest service location h'_t , $x = \|h_t - h'_t\|$ and the number of hops between the user and the service $d_t = \|\mu_t - h'_t\|$. Therefore, the cost in this time slot t is $M(x) + B(x) + T(y)$, where $d + e$ in (1) is h'_t , which gives y . However, there may be more than one shortest transmission path in the cell network between h_t and h'_t and any of them can be chosen. For convenience, the probability of a node being UAV-assisted on the transmission path is the same as the probability of the random distribution of UAVs in that small region.

As for service quality, the edge facility in which the user is located may be the best choice. However, there may be insufficient service capacity, excessive load and questionable security for the target edge server. Therefore, the above issues should also be considered as important factors before developing a migration decision. Our suggested method for calculating reliability value in edge nodes is also well suited to serve as a judging criterion. Services should be migrated to a location with higher reliability value and lower cost. Thus, we have the migration server selection instant reward function as

$$R = \iota T_n - \eta C, \quad (20)$$

where ι and η are weighting parameters. Further, C denotes the value in units of seconds.

The objective of the proposed mobility service migration approach is to maximizes long-term expected rewards from the initial state, which can be denoted by

$$V^*(s_0) = \max V_\pi(s_0), \quad \forall s_0 \quad (21)$$

where $V^*(s_0)$ is the long-term expected rewards generated by the strategy π and initial state s_0 . The maximisation problem can be solved utilizing the Bellman equation, which can be shown as

$$V^*(s_0) = \max_a \{R(s_0, a) + \gamma \sum_{s_1 \in L^*L} P(s_1|s_0, a)V^*(s_1)\}, \quad (22)$$

where $0 < \gamma < 1$ is the discount factor. $R(s_0, a)$ denotes the instant reward function when take the action a at the state s . In (21), $P(s_1|s_0, a)$ is the transition probability from s_0 to s_1 .

Deep Q network (DQN) [34] is one of the popular reinforcement learning methods. It uses deep neural networks to approximate state-action value $Q(s, a; \theta)$, where θ is the weight vector of deep neurons. So that it can find the optimal approximate action and solve (22). It is ideally suited to solve the problem of oversized state and action values in MDPs. Hence, in this work, we adopt the DQN approach to search for the optimal solution for our proposed complex 3-D service migration strategy. The pseudocode of our dynamic service migration algorithm based on DQN is shown in Algorithm 1.

Algorithm 1 3-D dynamic service migration strategy based DQN

Input: D -empty replay buffer; θ -initial network parameters; θ^- -copy of θ

Input: Possible locations and action set A

Output: Optimal actions for service migration

- 1: Initialize parameters θ and θ' in online network Q and target network Q' respectively, replay memory D and target Q network parameter update frequency p
- 2: **For** episode=1, M **do**
- 3: **Reset** environment: $s_0 = \{0, 0\}$
- 4: **For** t=1, T **do**
- 5: With probability ϵ **select** a random action a_t
- 6: Otherwise **select** $a_t = \max_a Q^*(s_t, a; \theta)$
- 7: Choose action a_t and observe immediate reward r_t
- 8: Update s_{t+1}
- 9: Store (s_t, a_t, r_t, s_{t+1}) in memory D
- 10: **Sample** random minibatch of (s_t, a_t, r_t, s_{t+1}) from D
- 11: **Calculate** the loss by $E[(r_t + \gamma \max_a Q(s_{t+1}, a; \theta') - Q(s_t, a; \theta))^2]$
- 12: **if** $t \% p == 0$; $\theta \rightarrow \theta'$
- 13: **End for**
- 14: **End for**

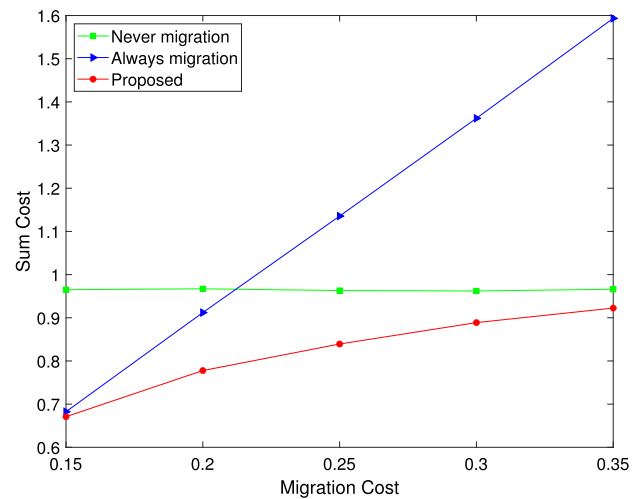
6 Numerical evaluation

In this section, we evaluate the performance of the proposed solutions under different parameter settings. We consider the following schemes first: 1) clients follow the random walk model with parameter $r = 0.15$ and maximum service range $N = 4$; 2) the UAV height is taken as the average value in the range; 3) in the blockchain system, the blocks generated have the same size and the frequency of generating blocks is the same.

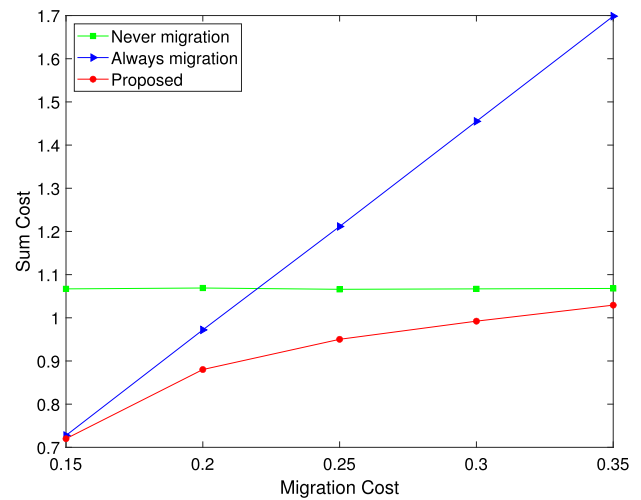
We first evaluate the impact of UAVs-assisted base station ratios and migration cost on service migration decisions. The discount factor γ of DQN is set as 0.9 and the batch size is set as 128. Further, the initial ϵ and final ϵ are set as 0.6 and 0.01, respectively. The transmission cost function parameters are selected as $\rho_c = -4$, $\rho_l = 4$, $\theta = 1.03$, where $\rho_c + \rho_l = 0$ means there is no constant portion in the cost function. Moreover, the migration is considered to be performed with the same and sufficient available bandwidth, the migration cost and security cost can therefore be equal to $(\vartheta + \delta)x$. Because of the complexity of the 3-D service migration environment, no other suitable strategy decision algorithm is available at this time. Therefore, the suggested solution is compared to some conventional migration strategies, including never migration and always migration, where always migration was proposed in [35]. Since the myopic method [12] results in the same results as never migration in the service range ($N < 5$), this paper does not additionally show its results. It is noteworthy that the experimental results are derived from the average of 20 randomly generated similar clients' walk routes in time slot $t = 6$.

Figure 4 demonstrates the comparison of migration cost versus sum cost for different migration policies. It is seen that the sum cost of always migration outperforms never migrate when the per migration cost is little. Nevertheless, as the block size grows or available bandwidth reduces, the sum cost of always migration increases exponentially following with the increase of per migration cost and surpasses the sum cost of never migrate. Interestingly, our proposed strategy achieves optimal results in various network conditions, which decreases the sum cost. Moreover, with the increasing proportion of the network's UAVs, the sum cost of our proposed solution gets closer to the cost of always migration, when the always migration has a lower cost than always transmission. That due to as the ratio of UAVs increases, the per transmission cost grows resulting in increased costs associated with the transmission when migrating services. Nevertheless, our proposed method still saves at most almost 20% of energy simultaneously over the baseline method with a 60% UAV ratio.

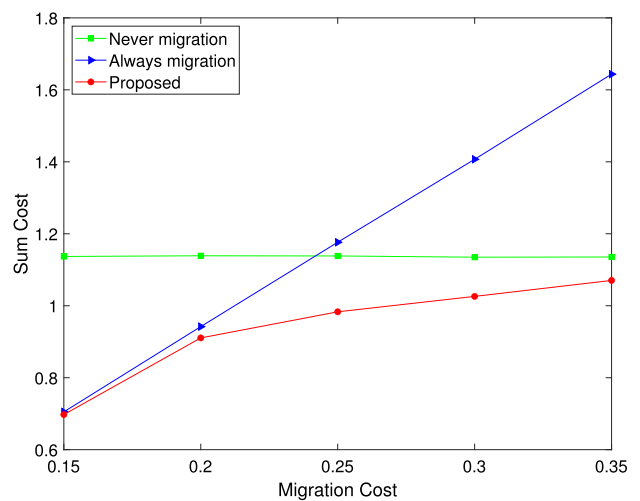
We then investigate the trust value evaluation scheme. We use the common MNIST dataset that was first introduced by LeCun et al. [36] and assume the training label is appropriate



(a)



(b)

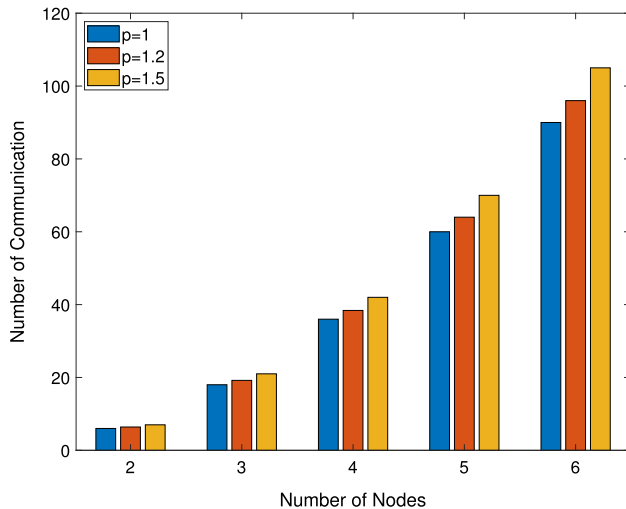


(c)

Fig. 4 Sum cost with different proportions of UAV in the network: **a** 0%, **b** 30%, **c** 60%

Table 2 Algorithm objective value evaluation

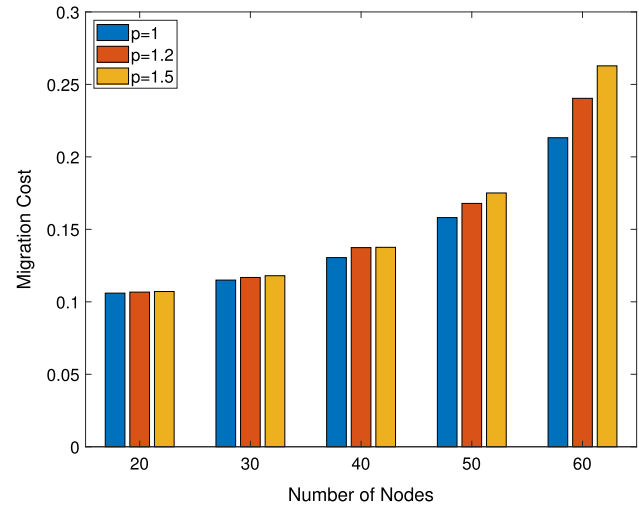
Hidden layer	Accuracy	Robustness	Neuron number	Complexity	Objective value
3	96.15%	20.7%	128/128/128	3.84	0.7643
4	96.62%	11.75%	128/.../128	5.12	0.5813
5	96.94%	6.58%	128/.../128	6.4	0.4082

**Fig. 5** Bandwidth consumption versus number of nodes

enough, i.e., $d = 1$. Since different deep learning algorithms have different complexity indicators, we set the DNN complexity indicator Ω to $\Omega = \sum_n^N \varphi/100$, where N is the hidden layer number and φ is the neuron number of each hidden layer. In addition, 100 is the number of reference neurons, which can be changed according to the model. The accuracy of the test set is shown in the table. Specifically, we used the Fast Gradient Sign Method (FGSM) [37] to construct attacks against DNN and thus derive algorithmic robustness. FGSM is one of the most straightforward and powerful adversarial attacks, which makes the DNN believe that the new input is safe by applying some small perturbations to the input instances without changing the training instances.

In Table 2, the objective trust value of the deep neural network (DNN) was utilised by DQN to demonstrate our reliability value evaluation scheme. As shown in Table 2, as the number of hidden layers increases, the accuracy and complexity increase but the robustness decrease, and therefore the objective value change. Thus, more accurate, faster, more robust and less complex algorithms can achieve higher value, which is in accordance with the expectations of a good algorithm. This provides a criterion for selecting the reliable intelligent autonomous node algorithm.

In Figs. 5 and 6, we illustrate how the number of nodes in the blockchain system affects the bandwidth and migration cost separately. We analyze the number of communications in

**Fig. 6** Migration delay versus number of nodes

the blockchain system to bandwidth consumption. Because the bandwidth consumption of a blockchain system increases exponentially as the number of nodes increases [38] and the consumption is 0 when the number of nodes is 1. Function (3) is suitable to liken the bandwidth consumption.

Figure 5 demonstrates that the bandwidth consumption grows with the number of nodes. It is seen that a lower probability of node conversion consistently enables a lower bandwidth consumption. The bandwidth consumption differential increases with the number of nodes, which directly affects the remaining available bandwidth per node.

Then in Fig. 6, we set the conversion parameter $b = 1$, the total bandwidth is 10 Mbps and nodes consume 500 bits per communication. As shown in Fig. 6, as the number of nodes increases, the migration latency increases slowly and then rapidly. This is because the increase in nodes leads to more bandwidth for security, resulting in increased migration costs and insufficient bandwidth significantly increases migration latency. Choosing the appropriate maximum service range N therefore also needs to be considered in a realistic scenario. Furthermore, the additional bandwidth consumption caused by under-ability or under-secured primary nodes increases the cost of migration. Similar to the trend seen in Fig. 6. Thus, reliability value is necessary to be considered in primary node selection to decrease p so that reduces the bandwidth and migration consumption.

7 Conclusion

In this paper, we proposed a blockchain-enabled energy-efficient secure migration framework in a 3-D communication environment where we studied the service migration decision optimisation problem. In particular, we presented a generalized intelligent autonomous node reliability calculation to improve system performance. To satisfy the client experience we modelled the service migration decision problem as an MDP and jointly optimised the cost of security, energy and service. A DRL algorithm was developed to handle the 3-D MDP problem, which can make nodes more reliable. Our results confirmed that intelligent autonomous node reliability effectively improves system performance and energy efficiency.

Acknowledgements This work was supported in part by the EU H2020 SANCUS project under the grant number GA952672.

Author Contributions GZ designed the proposed framework and perform the simulation. KN and QN wrote the main manuscript text. HP and CZ prepared figures 1–3 and the details of algorithm. All authors reviewed the manuscript.

Declarations

Conflict of interest The authors have not disclosed any conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ahmed, M. L., Iqbal, R., Karyotis, C., Palade, V., & Amin, S. A. (2022). Predicting the public adoption of connected and autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2), 1680–1688.
- Zheng, G., Ni, Q., Navaie, K., Pervaiz, H., & Zarakovitis, C. (2023). A distributed learning architecture for semantic communication in autonomous driving networks for task offloading. *IEEE Communication Magazine*, 6, 64–68.
- Tian, H., Xu, X., Qi, L., Zhang, X., Dou, W., Yu, S., & Ni, Q. (2021). CoPace: Edge computation offloading and caching for self-driving with deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 70(12), 13281–13293.
- Shinde, S., Bozorgchenani, A., Tarchi, D., & Ni, Q. (2021). On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems. *IEEE Transactions on Vehicular Technology*, 71, 2041–2057.
- Zhang, L., Wang, Z., & Zheng, G. (2023). OF-FSE: An efficient adaptive equalization for QAM-based UAV modulation systems. *Drones*, 7, 525.
- Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39.
- Ha, K., Abe, Y., Chen, Z., Hu, W., Amos, B., Pillai, P., & Satyanarayanan, M. (2015). Adaptive VM handoff across cloudlets, School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-15-113.
- Ranaweera, P., Jurcut, A. D., & Liyanage, M. (2021). Survey on multi-access edge computing security and privacy. *IEEE Communications Surveys, & Tutorials*, 23(2), 1078–1124.
- Ksentini, A., Taleb, T., & Chen, M. (2014). A Markov decision process-based service migration procedure for follow me cloud. In *2014 IEEE international conference on communications (ICC), 2014* (pp. 1350–1354).
- Wang, S., Urgaonkar, R., Zafer, M., He, T., Chan, K., & Leung, K. K. (2019). Dynamic service migration in mobile edge computing based on Markov decision process. *IEEE/ACM Transactions on Networking*, 27(3), 1272–1288.
- Satyanarayanan, M., Lewis, G., Morris, E., Simanta, S., Boleng, J., & Ha, K. (2013). The role of cloudlets in hostile environments. *IEEE Pervasive Computing*, 12(4), 40–49.
- Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of things. In *Proceedings of 1st edition MCC workshop mobile cloud computing, Helsinki, Finland* (pp. 13–16).
- Ning, Z., Chen, H., Ngai, E. C. H., Wang, X., Guo, L., & Liu, J. (2023). lightweight imitation learning for real-time cooperative service migration. *IEEE Transactions on Mobile Computing*.
- Wang, J., Hu, J., Min, G., Ni, Q., & El-Ghazawi, T. (2023). Online service migration in mobile edge with incomplete system information: A deep recurrent actor-critic learning approach. *IEEE Transactions on Mobile Computing*, 22(11), 6663–6675.
- Hu, J., Wang, G., Xu, X., & Lu, Y. (2019). Study on dynamic service migration strategy with energy optimization in mobile edge computing. *Mobile Information Systems*, 6, 1–12.
- Park, S. W., Boukerche, A., & Guan, S. (2020). A novel deep reinforcement learning based service migration model for mobile edge computing. In *2020 IEEE/ACM 24th international symposium on distributed simulation and real time applications (DS-RT)* (pp. 1–8).
- Zhang, C., & Zheng, Z. (2019). Task migration for mobile edge computing using deep reinforcement learning. *Future Generation Computer Systems*, 96, 111–118.
- Li, C., Zhu, L., Li, W., & Luo, Y. (2021). Joint edge caching and dynamic service migration in SDN based mobile edge computing. *Journal of Network and Computer Applications*, 177, 102966.
- Ciftcioglu, E. N., Chan, K. S., Urgaonkar, R., Wang, S., & He, T. (2015). Security-aware service migration for tactical mobile micro-clouds. In *MILCOM 2015-2015 IEEE Military communications conference* (pp. 1058–1063).
- Aslam, M., Bouget, S., & Raza, S. (2020). Security and trust preserving inter- and intra-cloud VM migrations. *International Journal of Network Management*, 6, 1–19.
- Cao, S., Wang, Y., & Xu, C. (2017). Service migrations in the cloud for mobile accesses: A reinforcement learning approach. In *2017 International conference on networking, architecture, and storage (NAS)* (pp. 1–10).
- Machen, A., Wang, S., Leung, K. K., Ko, B. J., & Salonidis, T. (2018). Live Service Migration in Mobile Edge Clouds. *IEEE Wireless Communications*, 25(1), 140–147.
- Ha, K., Abe, Y., Eiszler, T., Chen, Z., Hu, W., Amos, B., Upadhyaya, R., Pillai, P., & Satyanarayanan, M. (2017). You can teach elephants

- to dance: Agile VM handoff for edge computing. In *Proceedings of the second ACM/IEEE symposium on edge computing* (pp. 1–14).
24. Strunk, A., & Dargie, W. (2013). Does live migration of virtual machines cost energy? In *2013 IEEE 27th international conference on advanced information networking and applications (AINA)* (pp. 514–521).
 25. Dargie, W. (2014). Estimation of the cost of VM migration. In *2014 23rd International conference on computer communication and networks (ICCCN)* (pp. 1–8).
 26. Li, W., Feng, C., Zhang, L., Xu, H., Cao, B., & Imran, M. A. (2021). A scalable multi-layer PBFT consensus for blockchain. *IEEE Transactions on Parallel and Distributed Systems*, 32(5), 1146–1160.
 27. Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4), 352–375.
 28. Onireti, O., Zhang, L., & Imran, M. A. (2019). On the viable area of wireless practical byzantine fault tolerance (PBFT) blockchain networks. In *2019 IEEE global communications conference (GLOBECOM)* (pp. 1–6).
 29. Castro, M., & Liskov, B. (1999). Practical byzantine fault tolerance. In *Proceedings of 3rd symposium of system design and implementation*.
 30. Wang, J., Jing, X., Yan, Z., Fu, Y., Pedrycz, W., & Yang, L. T. (2020). A survey on trust evaluation based on machine learning. *ACM Computing Surveys*, 53(5), 1–36.
 31. Li, C., Guo, W., Sun, S. C., Al-Rubaye, S., & Tsourdos, A. (2020). Trustworthy deep learning in 6g-enabled mass autonomy: From concept to quality-of-trust key performance indicators. *IEEE Vehicular Technology Magazine*, 15(4), 112–121.
 32. Cao, S., Wang, Y., & Xu, C. (2017). Service migrations in the cloud for mobile accesses: A reinforcement learning approach. In *2017 International conference on networking, architecture, and storage (NAS)* (pp. 1–10).
 33. Buchman, E. (2016). Tendermint: Byzantine fault tolerance in the age of blockchains. Ph.D. dissertation, University of Guelph, Guelph.
 34. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with deep reinforcement learning. Technical report. Deepmind Technologies. [arXiv:1312.5602](https://arxiv.org/abs/1312.5602) [cs.LG]
 35. Ma, L., Yi, S., & Li, Q. (2017) Efficient service handoff across edge servers via docker container migration. In *Proceedings of the 2nd ACM/IEEE symposium on edge computing*.
 36. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2323.
 37. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *Proceedings of the 2015 international conference on learning representations*. Computational and Biological Learning Society.
 38. Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017). LSB: A lightweight scalable blockchain for IoT security and privacy. CoRR. [arXiv:1712.02969](https://arxiv.org/abs/1712.02969)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Guhan Zheng is a Ph.D. candidate in communication systems at the School of Computing and Communications, Lancaster University, Lancaster, U.K. His research interests include machine learning, semantic communication, dynamic communication networks and 3D communication systems.



Keivan Navaie is a Professor at the School of Computing and Communications, Lancaster University, Lancaster, U.K. His research interests include distributed cloud and edge computing, connectivity resilience in cyber-physical systems and IoT, applications of machine learning and artificial intelligence, and cognitive communications networks.



Qiang Ni is a Professor at the School of Computing and Communications, Lancaster University, Lancaster, U.K. His research areas include future generation communications and networking, including green communications/networking, millimeter-wave wireless, cognitive radio systems, 5G/6G, SDN, cloud networks, edge computing, dispersed computing, Internet of Things, cyber physical systems, artificial intelligence/machine learning, and vehicular networks. He has authored

or coauthored 300+ papers in these areas.



Haris Pervaiz is an Associate Professor at the School of Computer Science and Electronic Engineering, University of Essex, UK. His research interests include 5G & beyond, green communication, semantic communication, AI/ML based networking, mmWave communication, energy & backhaul energy efficiency and green Het-Nets. He is an associate editor for IEEE Access, Wiley ETT, Wiley ITL, IET Networks and IET Communications. He also organised various workshops and special sessions as lead workshop chair in various IEEE flagship conferences.



Charilaos Zarakovitis is a Senior Researcher at the Institute of Informatics and Telecommunications (MNL), National Centre for Scientific Research “Demokritos”, Greece. His research interests include quantum neural networks, machine and deep learning, green communications modelling, bioinspired and game-theoretic decision-making, cognitive radios, network virtualisation, statistical signal processing, and convex optimisation analysis.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com