Secure Social Networks in 5G Systems with Mobile Edge Computing, Caching, and Device-to-Device Communications

Ying He, F. Richard Yu, Nan Zhao, and Hongxi Yin

ABSTRACT

Mobile social networks (MSNs) have continuously been expanding and trying to be innovative. Recent advances of mobile edge computing (MEC), caching, and device-to-device (D2D) communications can have significant impacts on MSNs in 5G systems. In addition, the knowledge of social relationships among users is important in these new paradigms to improve the security and efficiency of MSNs. In this article, we present a social trust scheme that enhances the security of MSNs. When considering the trust-based MSNs with MEC, caching, and D2D, we apply a novel deep reinforcement learning approach to automatically make a decision for optimally allocating the network resources. Google TensorFlow is used to implement the proposed deep reinforcement learning approach. Simulation results with different network parameters are presented to show the effectiveness of the proposed scheme.

Introduction

Recently, with advances in wireless networks and mobile devices, mobile social networks (MSNs) have been developed rapidly to provide a variety of social services and applications to mobile users [1]. Millions of mobile users interact with each other to exchange information in MSNs, which will become one of the most important networking paradigms in the fifth generation (5G) wireless mobile systems [2].

In MSNs, the advances of wireless mobile networks can be taken to improve the performance of social networks. In recent years, the developments of wireless mobile networks have fueled a plethora of innovations in various areas, including mobile edge computing (MEC), content-centric networking (CCN), and device-to-device (D2D) communications, which will have significant impacts on MSNs. With MEC, computation resources are placed at the edge of wireless mobile networks in physical proximity to mobile users. Compared to traditional mobile cloud computing, MEC can provide faster interactive response by low-latency connections. Therefore, MEC has been envisioned as a promising technique to offer agile and ubiquitous computation augmenting mobile services and applications, including social services and applications [3].

Another promising technology in wireless mobile networks is content-centric networking (CCN). The basic principle behind CCN is to deliver content based on the interest in it instead of sending the conventional requesting message. A significant advantage of CCN is to provide native support for highly efficient content retrieval while enabling the enhanced capability for mobility. One of the key features of CCN is *in-network caching*, which can effectively reduce the duplicate content transmission in the network [4]. Recent studies of applying caching in MSNs show that traffic loads and latency can be significantly reduced in MSNs [2].

In addition, recent advances in D2D communications can be beneficial to MSNs as well [5]. With D2D communications, users in close proximity can directly communicate with each other via D2D links instead of accessing base stations (BSs) exclusively [6]. When it comes to content-centric MSNs, in spite of the smaller-sized storage (compared to that of BSs), the ubiquitous caching capability residing in mobile devices cannot be neglected due to their ubiquitous in-network distribution and ever increasing storage size.

Although some works have been done on applying recent advances of MEC, in-network caching, and D2D to improve the performance of MSNs, the *knowledge of social relationships* among users in MSNs is largely ignored in these new paradigms to improve the security and efficiency of data exchange, sharing, and delivery in MSNs. To fill this gap, in this article, we study trust-based secure social networks with recent advances in MEC, caching, and D2D. Considering the integrated network, the allocation of resources for subscribed users is complicated, especially when the conditions of the network resources are varying with time [7]. Therefore, we utilize a novel deep reinforcement learning approach to automatically achieve the resource allocation tasks. The distinct features of this article are as follows:

- We present a social trust scheme that enhances the security of MSNs. Unlike most existing schemes, we use *uncertain reasoning* to derive trust values due to the uncertainty in trust evaluation. The flexibility and elasticity of uncer-

This research was supported in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University (No. 2018D03), the Xinghai Scholars Program, the Fundamental Research Funds for the Central Universities under DUT17JC43, and Natural Sciences and Engineering Research Council of Canada (NSERC).

Ying He is with Dalian University of Technology and Carleton University; F. Richard Yu is with Carleton University; Nan Zhao (corresponding author) is with Dalian University of Technology and Southeast University; Hongxi Yin is with Dalian University of Technology.

Digital Object Identifier:
10.1109/MWC.2018.1700274

1536-1284/18/$25.00 © 2018 IEEE

It is required that users should be assured of other users' legitimacy and authenticity, that is, users are in reality what they claim to be on social networks. Indeed, authentication or access control is of extreme importance in social networks, and trust relationships naturally shaped through the social interations of users can help protect social networks.

tain reasoning make it successful in many fields, such as multi-agent systems, expert systems, and sensor data fusion [8].

- In this article, trust evaluation from direct observations and indirect observations is derived by using the Bayesian inference approach and Dempster-Shafer theory [9], respectively. Trust management is of great importance in MSNs. In [10], the authors worked on filtering trust opinions through reinforcement learning. They proposed an Actor-Critic Trust (ACT) model, which can significantly mitigate the adverse effect of biased testimonies.

We present a learning-based approach to achieve autonomous resource allocation without any manual intervention. Specifically, our approach is based on deep reinforcement learning, where an *agent* receives a set of observations from the integrated network, including the wireless channel conditions, each node's trust value, the contents in the cache, and the vacant computational capacity. Then the agent sends these parameters into the deep neural network and outputs the optimal actions. The resulting operator's revenue is observed and sent back to the agent as a reward. The agent trains and updates the deep neural network model based on the obtained reward. The process is iterated until the optimal actions are achieved.

The remainder of this article is outlined as follows. We describe the system, which includes MSNs with mobile edge computing, caching, and D2D communications. Next, the system model is presented. The social trust scheme with uncertain reasoning is presented. We formulate this system as a deep reinforcement learning problem. Simulation results are presented and discussed. Finally, we give the conclusion of the work.

System Description

In this section, we first present MSNs. Then mobile edge computing, in-network caching, and D2D communications are introduced to MSNs. Next, we present the proposed social trust scheme that enhances the security of MSNs.

Mobile Social Networks

With the ubiquitous coverage of wireless networks and popularity of mobile devices, MSNs have been developed rapidly to provide a variety of social services and applications to mobile users, focusing not only on the behavior but also on the social needs of the users [1]. Compared to conventional mobile wireless networks, where the client-server structure is the main networking paradigm for mobile users to obtain contents, mobile users in MSNs do not always contact remote servers to request contents. Instead, mobile users in MSNs can directly obtain contents from each other within a community based on their social ties. Within social communities, different mobile users have different interests in different contents.

Mobile Social Networks with Edge Computing, Caching, and D2D Communications

In MSNs, huge amounts of information-rich data will be exchanged by mobile users in a variety of social services and applications. Recently, cloud computing has been applied in diverse domains, including MSNs, where mobile user data needs to be transmitted to and processed in data centers. However, in practice, since data centers are usually far away from the mobile users, it is difficult for data centers to provide mobile users with low-latency services. In addition, it may not be economical or feasible to transmit a large amount of data from the mobile users to the data centers. To address these issues, MEC has recently been proposed, particularly for wireless mobile access networks, to bring computing resources closer to mobile users. There are two important characteristics of MEC: low latency and location awareness. MEC applies the concept of cloud computing in network edge nodes to facilitate services and applications, including mobile social services and applications.

In addition, due to user mobility and poor-quality wireless radio links, it is challenging to deliver huge amounts of data using the traditional client-server approach in MSNs [2]. Recent advances in caching can be extended to MSNs to address this issue. This innovative in-centric approach natively privileges the information (e.g., trusted information in a specific proximity of an event and a specific time period) rather than the node identity. In addition, with in-network caching, mobility and sporadic connectivity issues can be effectively addressed in MSNs.

Moreover, with D2D communications, users in close proximity can directly communicate with each other via D2D links instead of accessing BSs exclusively. As a promising approach to offload traffic from BSs, D2D communications can enable the sharing of radio connectivity and direct information delivery between two close users. By exploiting D2D communications, the content-centric social networks can benefit from the large number of mobile devices involved. That is because the integration of D2D into content-centric social networks can enable content caching not only in the air, but also in mobile devices [5].

Social Trust

The purpose of social networks is to enable people to interact with old acquaintances, and explore new relationships with other people as well. However, the proliferation of social networks attracts not merely faithful users, but also malicious users with adverse purposes. For example, an attacker can create a fake identity as a well-known figure to send out misleading or malicious information for some hidden purposes, or to profit from his/her high reputation. Therefore, it is required that users should be assured of other users' legitimacy and authenticity [11], that is, users are in reality what they claim to be on social networks. Indeed, authentication or access control is of extreme importance in social networks, and trust relationships naturally shaped through the social interations of users can help protect social networks.

There are two complementary classes of security approaches in MSNs: prevention-based (e.g., authentication and encryption) and detection-based (e.g., intrusion detection) approaches. Prevention-based approaches can effectively prevent misbehavior in networks. However, there are still chances that malicious nodes launch attacks in MSNs. Therefore, serving as the second wall of protection, detection-based approaches can

effectively help identify malicious misbehaviors. Whether prevention-based or detection-based security schemes, trust mechanisms play an important role, and trust relationships are the core information on which all security mechanisms are based [11].

An Example Use Case of Trust-Based Social Networks with MEC, Caching, and D2D Communications

One example in the case of social networks with MEC, caching, and D2D communications is illustrated as follows. Assume that a mobile user issues a video content request to its associated network. First of all, according to the description of the video content and the information about the user, the BS and other mobile users who can have D2D communications with the requesting mobile user will check whether or not its associated cache has the requested content. If yes, the cache will further examine to see if the version of its cached video content can be played and matches the mobile user. If still yes, the system will decide who (the BS or a mobile user) should directly send the requested content to the mobile user from the cache, taking into account the conditions of channels, trust values, energy consumptions, and so on. If no, the BS or a mobile user will extract the current video content and construct a computation task according to the size of the content involving the input data, codes, and parameters, as well as the number of CPU cycles needed to accomplish the computing/transcoding task. Then the system will decide who (the BS or a mobile user) should execute the computation, taking into account the compute capacity, conditions of channels, trust values, energy consumption, and so on. After the computation is finished, the BS or the mobile user sends the transcoded video content to the requesting mobile user. If the cache cannot find the matched video content, the BS has to retrieve the content from the Internet, and this will inevitably occupy some of the backhaul resources.

When we consider social trust with MEC, caching, and D2D communications in MSNs, traditional approaches have great difficulty in solving the optimization problem (i.e., making the optimal decision for allocating appropriate resources to users), especially when the available resources dynamically change with time.

In this article, we present a novel deep reinforcement learning approach to solve the optimization problem in trust-based social networks with MEC, caching, and D2D communications. Before we describe recent advances in deep reinforcement learning, we first present the social trust scheme with uncertain reasoning.

Social Trust Scheme with Uncertain Reasoning

We evaluate the trustworthiness of a mobile user by a real number Tr ranging from 0 to 1. In our model, the trust value Tr is jointly determined based on direct observations and indirect observations. The direct observation trust of a mobile user is defined as the estimated degree of trustworthiness from its directly connected mobile users based on their past experiences. However, the subjective evaluation from direct connections may be prejudiced; therefore, in order to be more objective and impartial, we also consider

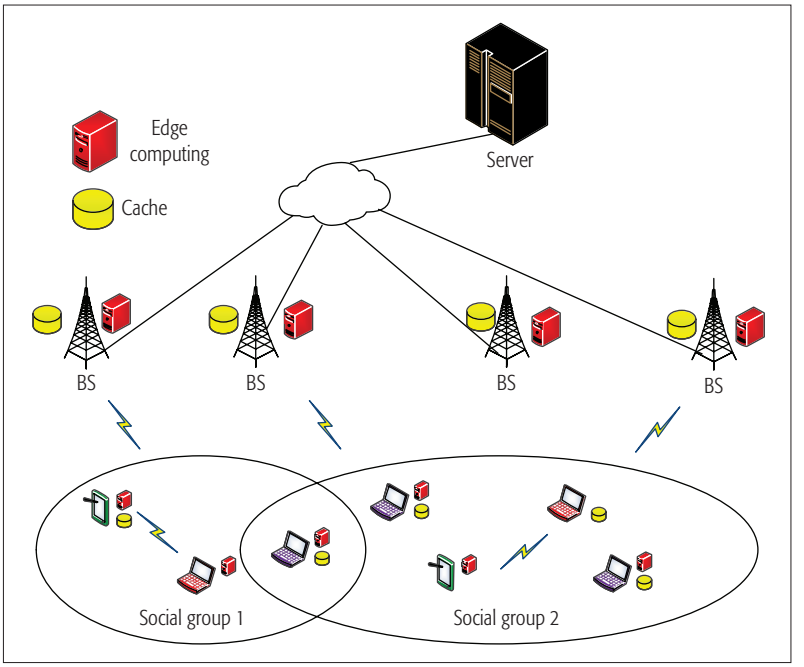

FIGURE 1. A mobile social network with edge computing, caching, and D2D communications.

the rating of trust from other indirectly connected mobile users. Here, we denote the trust value from direct observations as Tr^D and the trust value from indirect observations as Tr^{InD}. By combining these two trust values, we can obtain a more accurately estimated trust value of a mobile user as $Tr = \omega\, Tr^D + (1 - \omega)\, Tr^{InD}$, where ω is the weight coefficient to adjust the weightiness between direct and indirect observations, and $0 \le \omega \le 1$.

The trust evaluation procedure in our model can be visually explained as in Fig. 2. In the following, we discuss how to obtain the trust evaluation from direct observations and indirect observations by using the Bayesian inference approach and Dempster-Shafer theory [9], respectively.

Trust Evaluation from Direct Observations

In direct observations, consider that an observing mobile user can overhear the data forwarded by the observed mobile user and identify the observed mobile user's malicious behaviors, such as discarding or modifying some of the original data. Through multiple observations of the observed mobile user's behavior, the observing mobile user can evaluate the trust value by exploiting Bayesian inference, which is a method of statistical inference using Bayes' theorem to update the probability for a hypothesis when more evidence becomes available.

Under the Bayesian framework, we model the trust of a mobile user as a continuous random variable, denoted as Φ, where ϕ takes values from 0 to 1. We assume that Φ follows a beta distribution (i.e., $\Phi \sim Beta(a, b)$). Since Φ is assumed to obey a beta distribution, the trust value can be represented with the mathematical expectation of beta distribution as $\mathbb{E}_t[\Phi] = \{a_t\}/(a_t + b_t)$. Intuitively, the trust value of a mobile user is 0.5 at the beginning, and updated continuously by using the follow-up observations. For the detailed process of how to derive the trust values from direct observations, please refer to [12].

As is known, reinforcement learning is unstable or even divergent when a nonlinear function approximator such as a neural network is used to represent the Q-function. To address this instability, a biologically inspired mechanism called experience replay is utilized in deep Q-learning.

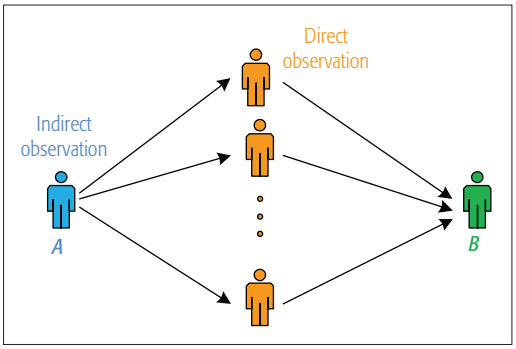

FIGURE 2. Social trust evaluation with both direct observation and indirect observation.

From the above discussion, we can see that past experiences play an important role in the Bayesian inference. In fact, recent behaviors of a mobile user should weigh more in the trust evaluation. Here, we introduce a punishment factor for reputation fading, which gives more weights on misbehaviors in the Bayesian inference. The trust evaluation formula is revised as follows: $\mathbb{E}_t[\Phi] = a_t/(a_t + \tau\, b_t)$, where τ is the punishment factor, and $\tau \geq 1$. With the increment of τ, the trust value declines quickly.

The punishment factor makes the trust evaluation more realistic and reliable. First, if a mobile user behaves maliciously once, compared to those who have no bad records, its trust value will be lowered much more. Second, the trust value will not recover quickly even if he/she behaves well because of the constraint of the punishment factor. This helps distinguish malicious mobile users quickly and prevent them disrupting others' trust evaluation. Based on the above deduction, the trust value from direct observation, Tr^D, is defined as: $Tr^D = \mathbb{E}_t[\Phi]$.

Trust Evaluation from Indirect Observations

Apart from direct observations, indirect observations from other mobile users are also very important in assessing the trustworthiness of an observed mobile user. Considering indirect observations helps mitigate the situation in which an observed mobile user is loyal to one mobile user but cheating on others. Assume that an observing mobile user collects observations from several other mobile users (also called a subsidiary observing mobile user), and combines the collected evidence into a decision about the observed mobile user's trust value. However, these subsidiary observing mobile users may be untrustworthy, or the evidence offered by them unreliable.

The Dempster-Shafer theory can be used as an effective way to handle the uncertainty issue and combine the evidence from multiple subsidiary observers. The core of this theory is based on two ideas: the degrees of belief about a proposition can be obtained from multiple subjective probabilities of a related theme, and these degrees of belief can be combined together under the condition that they are from independent evidence [9]. In the indirect observation, we assume that there are more than one subsidiary observing mobile users, and the evidence provided by them is mutually independent. Based on the definition of belief function, the Dempster-Shafer theory combines multiple users' belief.

Problem Formulation

In this section, we formulate an optimization problem of the secure social network with MEC, caching, and D2D communications. We assume that a mobile user requests a video content from the integrated network. For the network operator, it should decide which BS or D2D transmitter is assigned to serve the requesting user, whether or not video transcoding should be performed, and whether or not newly emerging contents should be cached. The network operator needs a comprehensive consideration of many factors, including the wireless channel conditions, whether or not the requested content is stored at the local cache, whether or not the content version is matched up, the computational capacity, and the trustworthiness of a D2D transmitter. Here, we consider dynamic scenarios, that is, the network conditions vary with time. We exploit the deep *Q*-learning algorithm to solve the formulated optimization problem.

We first give a brief description of the deep *Q*-learning algorithm. In order to obtain the optimal policy, identifying the system's states, actions, and reward functions is required, which is described in more detail below.

Deep *Q*-Learning

Q-learning is one of the most widely used model-free reinforcement learning methods. Recently, more advanced deep *Q*-learning is proposed, which approximates the *Q*-function with a deep neural network. Compared to traditional *Q*-learning, deep *Q*-learning utilizes two key ideas.

As is known, reinforcement learning is unstable or even divergent when a nonlinear function approximator such as a neural network is used to represent the *Q*-function. To address this instability, a biologically inspired mechanism called experience replay is utilized in deep *Q*-learning. As a matter of fact, the success of integrating reinforcement learning with deep neural networks was attributed to the incorporation of the experience replay technique that involves the storage and representation of recently experienced transitions [13].

To apply experience replay in deep *Q*-networks, we store the agent's experience tuple $e(t) = (x(t), a(t), r(t), x(t + 1))$ at each time step t into a replay memory $D(t) = \{e(1), ..., e(t)\}$. During learning, we uniformly and randomly draw mini-batches/samples from the replay memory to train the deep convolutional network's parameters. This approach exhibits several advantages. First of all, each experience tuple is possibly used in many parameter updates, which allows for greater data efficiency. Furthermore, learning directly from consecutive samples is inefficient because of the strong correlations between the samples. Thus, randomly sampling the data can remove the correlations in the observation sequences and smooth over the changes in the data distribution [13, 14].

The other key idea is that deep *Q*-learning adopts two separate convolutional networks to generate the target *Q* values and the estimated *Q* values. Since a deep *Q* network is trained toward the target value by minimizing the loss function, using one network for both the estimated *Q* val-

ues and target Q values would lead to falling into feedback loops. For the sake of stabilizing the training process, the target deep Q network's parameters are fixed and periodically updated.

System State and Action

The system state for a subscriber requesting video at a time slot mainly includes five components: channel state, computation capability, content indicator, version indicator, and the trust value for the video providers.

The system action includes which video provider is assigned to the subscribed user, whether or not the computation offloading (video transcoding) should be performed, and whether or not the video provider should cache the new video.

Reward Function

In this article, we set the system reward to be the total revenue of the network operator.

The network operator charges the subscribed user for associating with the video provider. On the condition that video transcoding (computation offloading) is decided to be executed on video provider's side, the operator can charge for its computing service. In addition, if the network controller decides to let the BSs cache the new video or new version, the operator gets a potential revenue on estimated backhaul savings.

On the other hand, the operator has to pay for the rented spectrum and backhaul bandwidth. Moreover, if video transcoding is performed, a certain amount of energy will be consumed for the computing that the network operator is obliged to pay. In addition, there is the cost of caching the video content in the memory.

The system reward for serving a subscribed user requesting a video is defined as the network operator's total revenue, and it is formulated as a function of the system states and actions. The system actions determine whether or not the reward can be optimized. The objective of adopting deep Q-learning into our network model is to help find an optimization policy that can maximize the accumulated future rewards for the network operator.

Simulation Results and Discussions

In this section, we evaluate the performance of proposed scheme using computer simulations. We use TensorFlow in our simulations to implement deep reinforcement learning. For performance comparison, the following four algorithms are presented:

1. The proposed scheme, which considers MEC, caching, and D2D, as well as both direct observation and indirect observation
2. The proposed scheme without indirect observation, which does not consider indirect observation
3. An existing scheme without mobile edge computing [15]
4. An existing scheme without D2D communications [2]

Simulation Settings

In the simulations, we consider an MSN consisting of 5 BSs, 5 MEC servers, and 15 D2D transmitters. The radius of the cell is set to 500 m. Because D2D communications typically perform within short ranges, we use a clustered-based distribution model, where multiple users are located within one cluster with a radius of 50 m. In the network, there are 20 subchannels, each of which has a bandwidth of 180 kHz. The transmit powers of a D2D transmitter and BS are 24 dBm and 46 dBm, respectively. The noise spectral density is −174 dBm/Hz. A loss model 35.3 + 37.6 log ($d(m)$) is used. In addition, block fading with a block size of 100 is assumed for channel fad-

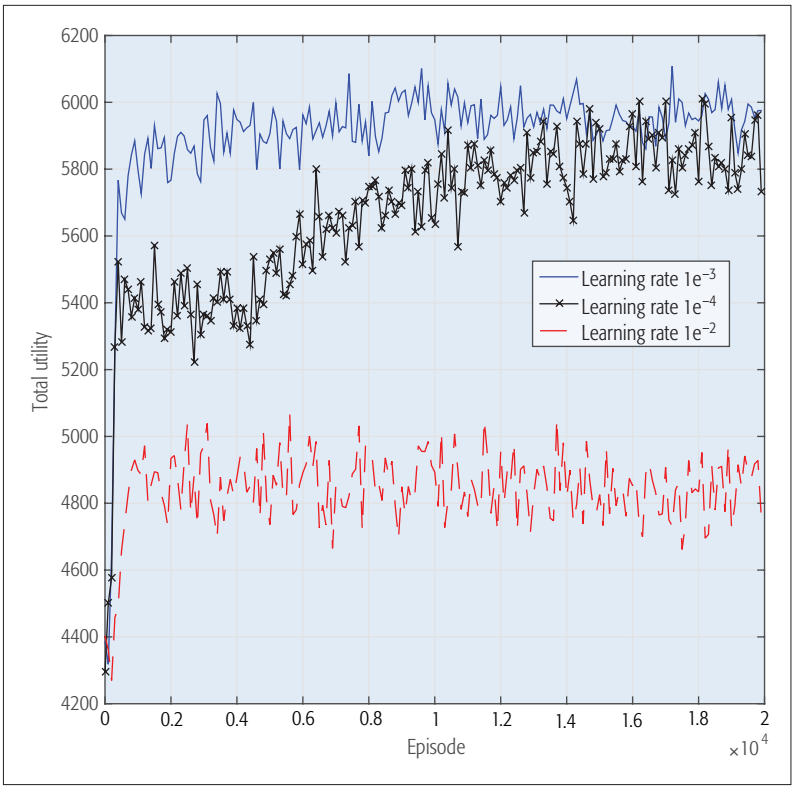

FIGURE 3. Convergence performance with different learning rates.

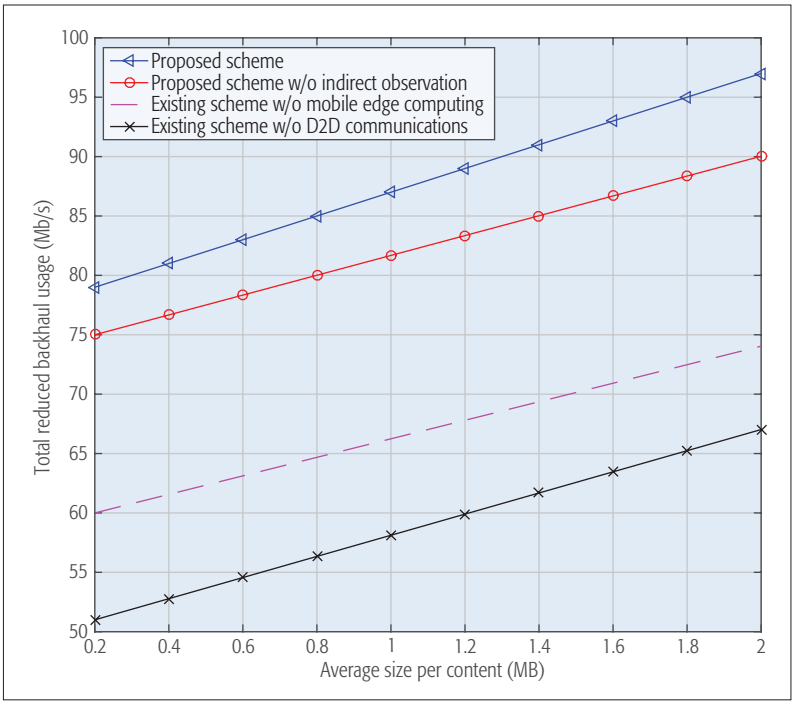

FIGURE 4. The total reduced backhaul usage vs. average size per content.

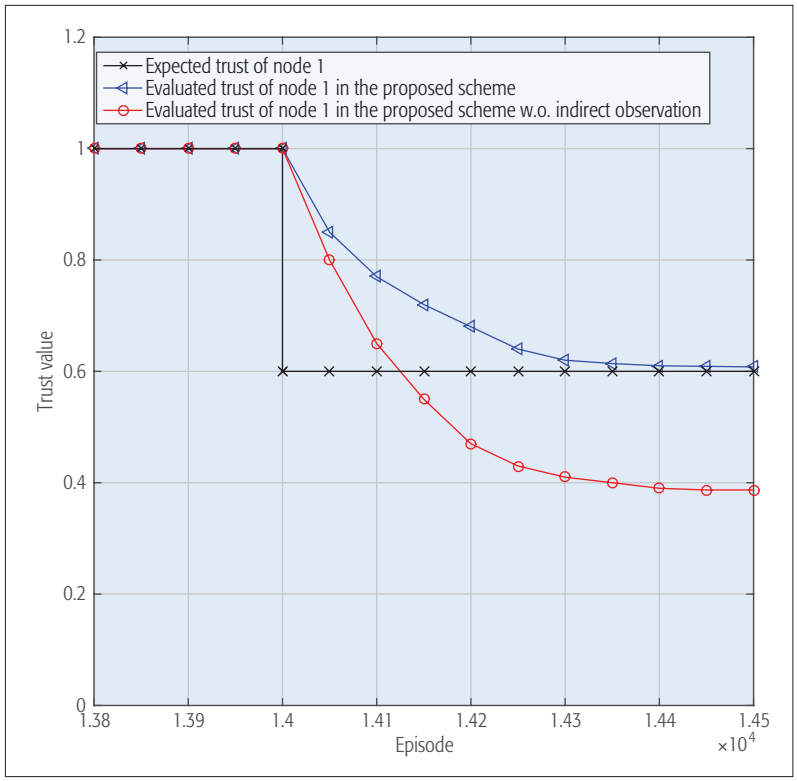

FIGURE 5. Total utility vs. episodes.

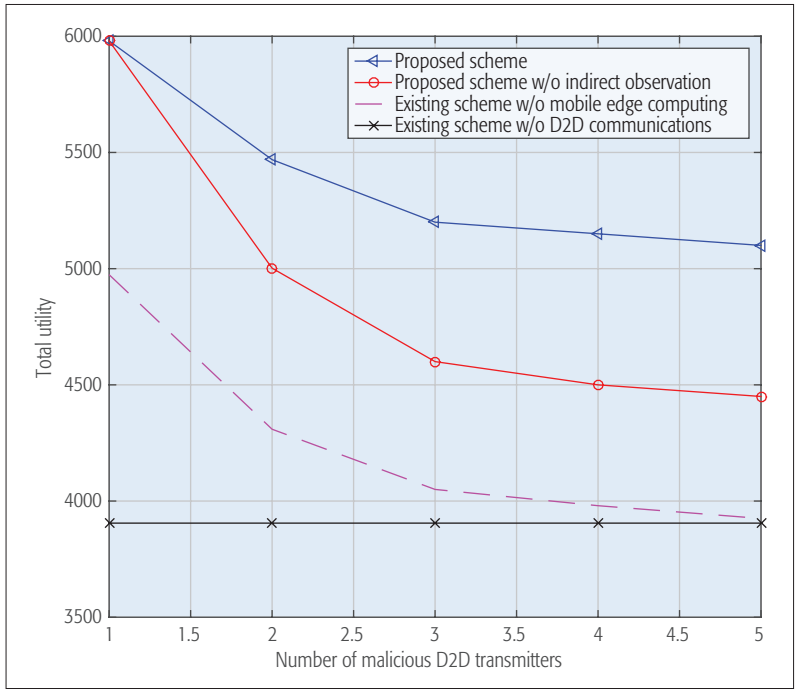

FIGURE 6. Total utility vs. the number of malicious D2D transmitters.

ing. Moreover, we assume that there are totally 10 types of contents distributed in the network, and each content cache state follows the Markov model. We set the cache state transition probability of staying in the same state in the BS as 0.6 (0.3 in the D2D transmitter), and set the transition probability from one state to another as 0.4 (0.7 in the D2D transmitter). We further assume a Zipf popularity distribution in the simulations, with θ = 1.5. The computation states of MEC servers follow the Markov model. We assume that the computation state of the MEC server can be very low, low, medium, high, and very high.

We assume that there are two types of D2D transmitters in the network: normal nodes, which share the content and perform computing normally, and compromised nodes, which modify contents maliciously. The BSs are assumed to be not compromised due to the physical security of BSs. We also assume that the number of compromised nodes is much lower than the total number of nodes in the network. The attackers are independent. Hence, there is no collusion attack in the network.

Simulation Results

The convergence performance of the proposed scheme using the deep reinforcement learning algorithm is shown in Fig. 3. We can observe that at the beginning of the learning process, the total utility of the proposed scheme is very low. As the number of the episodes increases, the total utility increases until it converges to a relatively stable value, which is around 5900. We can also observe the convergence performance of the proposed scheme with different learning rates. For example, the convergence of the proposed scheme is faster when the learning rate is 0.001 compared to the case when the learning rate is 0.0001. However, this does not mean that a larger learning rate is always better, because a larger learning rate may result in local optimum just as the algorithm converges to the point around 4900 with the learning rate of 0.01. Therefore, it is important to choose an appropriate learning rate for a specific problem. For our studied problem, we choose the learning rate of 0.001 in the rest of the simulations.

The effects of average size per content on the total reduced backhaul usage is shown in Fig. 4. We can see that the total reduced backhaul usage increases with the increase of the average size per content. Compared to the existing schemes without mobile edge computing and D2D communications, the proposed scheme has larger total reduced backhaul usage due to the benefits of MEC and D2D communications. In addition, without indirect observation, the accuracy of trust evaluation is lower in the proposed scheme, which induces a lower gain of MEC and caching, and lower total reduced backhaul usage.

Next, we study the performance of the proposed social trust scheme. Assume that at episode 14K, D2D transmitter 1 changes its maliciousness to 0.6. Our goal is to observe the accuracy of trust tracking in this scenario where the malicious behavior of a D2D transmitter changes rapidly. Figure 5 shows the trust tracking of the system using direct observation with Bayesian inference and the proposed scheme using both direct observation with Bayesian inference and indirect observation with the Dempster-Shafer theory. We can observe from Fig. 5 that only direct observation can result in inaccurate trust values. In contrast, the proposed scheme can track the trust value accurately with both direct observation and indirect observation.

The number of malicious D2D transmitters in the network also has a significant impact on the network performance. Here, we investigate the

system utility with the number of malicious D2D transmitters, from 1 to 5, in a 15-D2D-transmitter environment. The basic parameter is the same as above. Figure 6 shows that as the number of malicious D2D transmitters increases, the utility drops dramatically. When the number of malicious D2D transmitters reaches one third of the total number of D2D transmitters in the network, the utility decreases to about half of the utility in the network with one malicious node. From this figure, we can see that the network is deeply affected by the number of malicious D2D transmitters.

Conclusions and Future Work

Mobile social networks have become one of the most important networking paradigms in future wireless mobile networks. In this article, we study the impacts of recent advances of mobile edge computing, content-centric networking, and D2D communications on mobile social networks. In addition, we consider the knowledge of social relationships in these new paradigms to improve the security and efficiency of mobile social networks. Specifically, we present a social trust scheme with both direct observation using Bayesian inference and indirect observation using the Dempster-Shafer theory. We further propose a deep reinforcement learning approach to study this complicated system. Extensive simulation results are presented to show the effectiveness of the proposed scheme. Future work is in progress to consider using blockchain technology in the proposed framework.

References

[1] N. Vastardis and K. Yang, "Mobile Social Networks: Architectures, Social Properties, and Key Research Challenges," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 3, 2013, pp. 1355–71.

[2] Z. Su and Q. Xu, "Content Distribution over Content Centric Mobile Social Networks in 5G," *IEEE Commun. Mag.*, vol. 53, no. 6, June 2015, pp. 66–72.

[3] Y. He *et al.*, "Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," *IEEE Commun. Mag.*, vol. 55, no. 12, Dec. 2017, pp. 31–37.

[4] C. Liang *et al.*, "Enhancing QoE-Aware Wireless Edge Caching with Software-Defined Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 16, Oct. 2017, pp. 6912–25.

[5] X. Wang *et al.*, "Tag-Assisted Social-Aware Opportunistic Device-to-Device Sharing for Traffic Offloading in Mobile Social Networks," *IEEE Wireless Commun.*, vol. 23, Aug. 2016, pp. 60–67.

[6] K. Wang, F. R. Yu, H. Li, and Z. Li, "Information-Centric Wireless Networks with Virtualization and D2D Communications," *IEEE Wireless Commun.*, vol. 24, June 2017, pp. 104–11.

[7] Y. He *et al.*, "Big Data Analytics in Mobile Cellular Networks," *IEEE Access*, vol. 4, Mar. 2016, pp. 1985–96.

[8] F. Yu *et al.*, "A First-Order Logic Framework of Major Choosing Decision Making with an Uncertain Reasoning Function," *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 2017.

[9] T. M. Chen and V. Venkataramanan, "Dempster-Shafer Theory for Intrusion Detection in Ad Hoc Networks," *IEEE Internet Comp.*, vol. 9, no. 6, Nov. 2005, pp. 35–41.

[10] H. Yu *et al.*, "Filtering Trust Opinions Through Reinforcement Learning," *Decision Support Systems*, vol. 66, Oct. 2014, pp. 102–13.

[11] C. Zhang *et al.*, "Privacy and Security for Online Social Networks: Challenges and Opportunities," *IEEE Network*, vol. 24, no. 4, July/Aug. 2010.

[12] R. Changiz *et al.*, "Trust Establishment in Cooperative Wireless Networks," *Proc. IEEE MILCOM '10*, Nov. 2010, pp. 1074–79.

[13] V. Mnih *et al.*, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, Feb. 2015, pp. 529–33.

[14] Y. He *et al.*, "Deep Reinforcement Learning-Based Optimization for Cache-Enabled Opportunistic Interference Alignment Wireless Networks," *IEEE Trans. Vehic. Tech.*, vol. 66, Nov. 2017, pp. 10433–45.

[15] Y. Zhao, W. Song, and Z. Han, "Social-Aware Data Dissemination Via Device-to-Device Communications: Fusing Social and Mobile Networks with Incentive Constraints," *IEEE Trans. Services Computing*, 2017.

Biographies

Ying He [S'16] (heying@sce.carleton.ca) received her B.S. degree from Dalian Ocean University, China, and her M.S. degree from Dalian University of Technology in 2011 and 2015, respectively, both in communication and information systems. She is currently pursuing a Ph.D. degree with both Dalian University of Technology and Carleton University. Her current research interests include machine learning, security, big data, and wireless networks.

F. Richard Yu [S'00, M'04, SM'08, F'18] (richard.yu@carleton.ca) is a professor at Carleton University, Canada. His research interests include connected vehicles, security, and wireless communications. He serves on the Editorial Boards of several journals, including Co-Editor-in-Chief for *Ad Hoc & Sensor Wireless Networks*, Lead Series Editor for *IEEE Transactions on Vehicular Technology*, and *IEEE Transactions on Green Communications and Networking* and *IEEE Communications Surveys & Tutorials*. He is a Distinguished Lecturer and Vice President (Membership) of the IEEE Vehicular Technology Society.

Nan Zhao [S'08, M'11, SM'16] (zhaonan@dlut.edu.cn) is an associate professor at Dalian University of Technology. He received his B.S. degree in electronics and information engineering in 2005, his M.E. degree in signal and information processing in 2007, and his Ph.D. degree in information and communication engineering in 2011, all from Harbin Institute of Technology, China. His research interests include interference alignment, cognitive radio, green communications, and physical layer security.

Hongxi Yin (hxyin@dlut.edu.cn) is a professor at Dalian University of Technology. He received his B.S. degree from Shandong University in 1982, his M.E. degree from Harbin Institute of Technology in 1988, and his Ph.D. degree from Zhongshan University, China, in 1998. He worked as a postdoctoral researcher and an associate professor at Peking University, China, from 1998 to 2008, and a research fellow at the University of Southampton, United Kingdom, from 2005 to 2007.

The system reward for serving a subscribed user requesting a video is defined as the network operator's total revenue, and it is formulated as a function of the system states, and actions. The system actions determine whether or not the reward can be optimized. The objective of adopting deep Q-learning into our network model is to help find an optimization policy that can maximize the accumulated future rewards for the network operator.