Please cite the Published Version

Leveraging Graph Convolutional-LSTM for Energy Efficient Caching in Blockchain-based Green IoT

Ge Chen, Jun Wu, *Member, IEEE,* Wu Yang, Ali Kashif Bashir, *Senior Member, IEEE,* Gaolei Li, and Mohammad Hammoudeh, *Senior Member, IEEE*

Abstract—Nowadays, adopting blockchain technology to Internet of Things has become a trend and it is important to minimize energy consumption while providing a high quality of service (QoS) in Blockchain-based IoT networks. Pre-caching popular and fresh IoT content avoids activating sensors frequently, thus effectively reducing network energy consumption. However, the user equipment in regions covered by base stations will generate distributed and time-varying data requests, hence modeling the base station topology to capturing spatio-temporal request patterns is required for the data storage pre-allocation. Traditional solutions typically fail to pay attention to the topology, resulting in the sensor being activated redundantly. In this paper, we propose Request Graph Convolutional-LSTM to capture the spatio-temporal request patterns in Blockchain-based IoT networks and make predictions. Moreover, a heuristic algorithm based on the predictions is proposed to develop pre-caching strategy, which determines the data and location to be cached to minimize the mean data retrieval latency restricted by the cache space of IoT network entities and the freshness of IoT content. Experiments show that our proposed frame provides a low energy consumption.

Index Terms—Internet of Things, Graph convolution networks, Blockchain, spatial-temporal, pre-caching.

I. INTRODUCTION

INTERNET of Things (IoT) continues integrating with human society in the 5G era. It is anticipated that by the year 2022, 18 billion devices will be connected to the IoT [1]. Blockchain technology, which can establish a secure, credible, and decentralized data sharing ecosystem, has attracted widespread attention from researchers. The blockchain [2] consisting of a set of blocks and transactions is a distributed database that eliminates the need for a central authority. Blockchain has become an important component of the fundamental elements of in Industrial Internet of Things (IIOT) [3], Vehicle-to-everything (V2X) [4], and IoT because of its security, decentralization, and autonomy.

Ge Chen, Jun Wu, and Gaolei Li are with Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China. (e-mail: junwuhn@sjtu.edu.cn)

Wu Yang is with the Information Security Research Center, Harbin Engineering University, Harbin 150001, China. (e-mail: yangwu@hrbeu.edu.cn)

Ali Kashif Bashir is with Department of Computing and Mathematics, Manchester Metropolitan University, UK, and School of Electrical Engineering and Computer Science, National University of Science and Technology (NUST), Islamabad, Pakistan.

Mohammad Hammoudeh is with Department of Computing and Mathematics, Manchester Metropolitan University, UK.

In blockchain-based IoT, local edge nodes are essential for building a decentralized storage system. Edge (or fog) computing enhances the computing and storage resources of local edge nodes. Artificial intelligence (AI) services empowered by edge computing are gradually being pushed to the edge node hence also known as edge artificial intelligence (edge-AI) [5]. In contrast to traditional centralized storage, storing on the blockchain utilizes more advanced redundant coding modes to store data on tens of millions of nodes around the world. Transmitting all the data to the data center requires huge bandwidth, which will eventually generate huge energy consumption. Meanwhile, there will be data security concerns. With the adoption of novel methods to reduce energy costs, blockchain-based storage will have more advantages. Sensing service, as the primary application of IoT, transmits the sensing data and assists users in monitoring the environmental conditions (e.g., traffic congestion, humidity, and temperature) [6]. With the number of users and IoT sensors growing rapidly, a considerable amount of traffic will flood into the IoT network in the near future. On the other hand, frequent requests for data transmission from IoT sensors accelerate battery power consumption [7]. Generally, sensors have limited battery capacity and are equipped with batteries with limited energy. An effective technology that can avoid activating sensors frequently (consuming substantial energy) is to cache IoT data at network entities [8]. Attaching cache to IoT gateways (GW) was proposed in [9]. For time-varying and dynamic IoT networks, the authors utilized deep reinforcement learning to manage content placement. Another caching strategy adopting markov chain model to allocate edge cache resources reasonably in blockchain-based IoT was proposed in [10]. To sum up, developing an efficient caching strategy on network entities is essential to achieve the reduction of sensor energy consumption.

However, numerous base stations (BSs) are geographically distributed on a large scale in cities, and mobile users usually stay briefly and quickly switch between the areas covered by different BSs. Thus, users' requests for sensing data demonstrate heterogeneity (Spatio-temporal and relation). Designing an energy-efficient caching strategy should realize the goals as follows.

1) The number of sensor activations should be minimized to reduce energy consumption.

2) The cache should be reasonable deployed near the edge of the network to effectively reduce repeated transmissions.

3) Considering the geographical distribution of BSs, the content replacement problem should be solved by predicting

the requested sensing data based on capturing the relationship between the coverage areas of BSs, rather than just based on the popularity of the data.

Recently, deep learning on graph-structured data has become a topic that grabs widespread attention in academia. With the support of the spectral graph theory [11], a graph convolution based on the Laplacian matrix [12] is designed to deal with the inherent combinatorial complexity of the graph structure. To lessen the complexity of learning, the local spectrogram convolution [13] is also proposed. The graph convolution neural networks based on adjacency matrix [12], [14] is more flexible because of the incorporation of the adjacency matrix. The distribution of BSs in the IoT can be considered as a general graph structure, on which the convolution operator can be extended to capture the spatial relationship. Long short-term memory neural network (LSTM) [15] is often used in time series forecasting and analysis tasks, i.e., utilizing the characteristics of an event in the past period of time to predict its future characteristics. LSTM can be applied to capture the temporal characteristics of the sensing data requested by users.

In this paper, motivated by previous works, we are dedicated to proposing a deep learning architecture, which incorporates the graph convolutional network and LSTM into blockchain-based green IoT for pre-caching, to decrease energy consumption.

This main contribution is to include the following three aspects:

- A deep learning architecture simultaneously capturing the unstable temporal dynamics and spatial dependencies in blockchain-based green IoT that reduces the energy consumption is proposed.
- Different from traditional caching strategies based on popularity, Request Graph Convolution (RGC) with three convolutional layers which can learn the characteristics of localized data requests is designed for making accurate predictions. According to the sensing data requests predictions, a low-complexity pre-caching heuristic algorithm is designed to decrease energy consumption in the blockchain-based IoT.
- Extensive simulations on synthetical dataset indicate that our framework is superior to traditional approaches in the aspect of effectiveness and energy consumption.

The remainder of the paper is summarized as follows: In Section II, we summarize the related works. In Section III, we present a comprehensive explanation of our system model. Section IV describes leveraging graph convolutional network and LSTM for interesting data prediction. A low-complexity heuristic pre-caching algorithm is developed in Section V. Section VI shows the experiment results. The conclusion of our paper is presented in Section VII.

II. RELATED WORK

There are numerous studies that investigate energy saving in IoT networks. In [16], the authors illustrated that the energy consumed by IoT can be significantly diminished through caching IoT resources. Quevedo et al. [17] analyzed that

caching in-network contributes to reducing both bandwidth usage as well as energy consumption. On the other hand, many researchers also focused on adopting blockchain technology equipped with a set of good characteristics (i.e., decentralization, security, etc.) for IoT support. In [18], the authors gave a brief description of challenges posed by the high dynamics of IoT applications. To dynamically configure the blockchains, a software-defined blockchain structure is designed. Moreover, a consensus function virtualization method and a transfer-learning-based intelligent project are proposed to enhance the flexibility and scalability of blockchains in IoT.

Several studies focused on designing efficient caching strategies. In [9], to reduce the energy consumed by sensors, the authors proposed that the cache in GW can be leveraged to temporarily place popular sensing data. Hence, the number of activations of the sensor is reduced. Deep reinforcement learning is utilized to determine content placement within dynamic IoT networks. Cloud radio access network (C-RAN), as a patent framework of 5G networks, aggregates the computational functionalities of conventional BSs in a centralized resource pool [20], namely baseband unit (BBU) pool. Thus, their work ignores the BBU buffer space, and the network architecture they proposed is overly simplified. In [19], a collaborative caching scheme for IoT in 5G networks was proposed, which provides space-reserved storage at both small and macro BSs to decrease the average energy consumption. However, their work did not take take into account the impact of user mobility of the IoT on data storage. In [10], the authors proposed to utilize the caching capacity of edge server to support the IoT devices in caching data in blockchain-based IoT. And there is a shortcoming that the above work has not considered the geographical distribution of base stations and the spatial-temporal characteristic of sensing data. In our work, we analyze the spatial-temporal IoT networks where the influence of both time and geographic factors on sensing data requests is considered. In this setting, we study the data pre-caching issue in blockchain-based IoT networks by leveraging graph convolutional-LSTM.

Studies that investigate conducting graph convolution are mainly divided into two classes. The first class of methods [12], [13], [21] started from the graph Laplacian matrix and designed a spectral filter convolution. The other form of graph convolution is to dynamically perform on graph data structure. An example was proposed in [22] when performing graph convolution, the dynamic edge-conditioned filters are applied. LSTM Encoder-Decoder model for data caching was first proposed in [23], and can significantly boost cache efficiency. To design reasonable caching strategies, LSTM is leveraged as a forecasting tool to forecast the data popularity [24].

III. SYSTEM MODEL

As shown in Fig.1, the system model of our proposed blockchain-based deep hierarchical pre-caching architecture with BBU pools and IoT GW comprises several planes: the user perception plane, distributed edge artificial intelligence plane, and IoT sensor plane.

The user perception plane consisting of BSs geographically distributed in the city can monitor sensing data requests
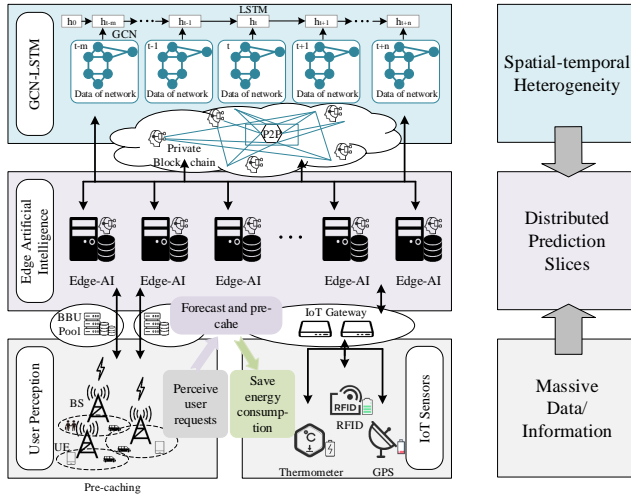
Fig. 1. System Model.

generated from the user equipment (UE). Near the edge of the network, massive information is periodically transmitted to edge-AIs within the edge artificial intelligence plane for processing. The graph convolution network (GCN) and LSTM relying on certain computing capability, are deployed in the distributed edge-AIs to simultaneously learn the complicated spatial dependency and temporal dynamics in green IoT. Therefore, edge-AIs, also acting as an edge data center, performing calculations, and ultimately generating sensing data request predictions. The underlying infrastructure in IoT (e.g., thermometer, GPS, etc), which sense the environment, is deployed in the sensor plane. Massive IoT data (e.g., temperature, traffic conditions, etc) is generated at this layer and transmitted to the consumers who request them.

It is unrealistic and inefficient to manage edge-AIs centrally due to the geographical distribution of them. Additionally, edge-AIs opens up the possibility of attacks from malicious attackers, which may result in the collapse of the entire network. The sensing data request predictions are stored in edge-AIs and could be attacked by adversaries. Malicious attackers could tamper with the prediction results in the victim edge-AIs. The integrity and non-tampering of prediction results should be guaranteed. Therefore, decentralized management of edge-AIs is essential to guarantee the security and effectiveness of our architecture.

Blockchain technology has been utilized to promote trust and data privacy while maintaining trust, integrity, and transparency. The blockchain technology supporting P2P communication is utilized to provide to guarantee effective and reliable IoT network management. It provides identity verification services for IoT entities, carries management responsibilities, and grants access to data (i.e., access control). However, for (open) public blockchains, a large number of edge-AIs make transaction verification inefficient. Moreover, the mining processes result in higher energy consumption and long delays. Thus, private blockchain requiring less energy is utilized. Private blockchains can provide a high level of security, which guarantees a secure ecosystem for data sharing. In

the proposed framework, the private blockchain is deployed among the edge-AIs. The private blockchain will not expose transactions to the entire network due to its particularity. Entities involved in transactions also have a high degree of trust. Therefore, the private blockchain guarantees both the privacy and transaction efficiency of edge-AIs. Besides, applying encryption helps minimize undesired access to the stream under the circumstance of data sharing.

According to the system architecture, edge-AIs, being deployed on the edge computing platform, have the advantages of low latency, energy-saving, and reduction of core network congestion. The algorithm we proposed is deployed on edge-AIs and predicts the data requests in each region accurately, so reasonable storage allocation is performed hierarchically on the BBU pools and GWs. Therefore, frequent sensor activation can be avoided, which saves considerable energy.

In the following, we apply a request graph convolution operator to capture the spatial dependencies of data requests between regions covered by BSs. Base on this operator, we predict data requests from UEs and propose a pre-caching strategy.

IV. INTERESTING DATA PREDICTION BASED ON REQUEST GRAPH CONVOLUTIONAL-LSTM

Normally, a graph is a sort of network data structure. A graph G is composed of nodes set V and edge set E. The relationship between nodes is called an edge. We consider an active base station distribution map in the city of Ghent [25]. Each base station covers a circular area, and some areas are adjacent. The graph representing a network of regions covered by the BSs distinct from biomolecule network graphs or paper reference graphs in several aspects: 1) the structure of the network rarely changes; 2) the status of requests for interesting IoT data in each region changes over time; 3) regions in the network graph possess meaningful physical properties, such as the coverage area, etc.

A. Network Topology

The map network is described as an undirected graph $B(A, J)$ based on geographic distributed BSs, where the set of nodes A represents the areas covered by the base station, and the edge set J represents the adjacent areas.

1) Adjacency Matrix: The information of the edges or arcs of vertices in A is portrayed by an adjacency matrix J, and the pseudo-variable $J(i, j)$ illustrates whether area i and area j are interconnected, i.e., each item $J_{i,j} = 1$ if there exists an edge linking vertice i and vertice j together or $J_{i,j} = 0$ ($J_{i,i} = 0$). However, since the content popularity is time-series-based data. The current status of an area affects the status in the future. Moreover, we deem that all areas are self-influencing. Therefore, we deem that the neighborhood of a vertice not only includes itself but a matrix describing the one-order adjacency relation of the entire graph, expressed as $\tilde{J} = J + I$, I of which is the unit matrix.

2) Spectral Graph Convolution: The idea of graph convolution is to extend standard convolution to general topology graphs by operating in the spectral field. Supported by the Laplacian matrix, the spectral graph convolution is established in the Fourier domain [11], which equates $L = D - J$, D of which is the degree matrix. Laplacian matrix can be diagonalized as $L = U\Lambda U_T$ according to eigendecomposition due to its positive semi-definite property. Characteristic values compose a diagonal matrix Λ, eigen vectors form U, and the transposed matrix of U is denoted as U_T. Multiply a signal $x \in R_N$ with a filter $g_\theta = diag(\theta)$ $(\theta \in R_N)$ defines the spectral convolutions operation on topological graphs in the Fourier domain [26], i.e.:

$$g_\theta \star x = U g_\theta U_T x, \tag{1}$$

where g_θ is a function of the characteristic value of L, i.e. $g_\theta(\Lambda)$. Furthermore, for saving computational cost, it was suggested [27] to fit the convolution kernel with Chebyshev polynomials:

$$g_{\theta'} = \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda}), \tag{2}$$

where $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$ is the diagonal matrix of re-scaled eigenvalues, $\theta' \in \mathbb{R}^K$ is Chebyshev coefficients (i.e., parameters updated iteratively in training). Chebyshev polynomial does not affect matrix operations, thus matrix operation is performed first. The convolution after substitution is expressed as:

$$g_{\theta'} \star x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x, \tag{3}$$

where $\tilde{L} = \frac{2}{\lambda_{max}}L - I_N$ depends on the node-set with the largest K step distance from the center node. Eigenvector decomposition is not required during the calculation process.

B. Data Request Rate Prediction Problem

TABLE I
NOTATIONS FOR DATA REQUEST RATE PREDICTION

Notations	Description
B	Request region-based graph network $B = (A, J)$.
J	Set of edges in B possessing the the number of elements $|J|$.
A	Set of vertices in B possessing the the number of elements $|A| = N$.
$J \in \mathbb{R}^{N \times N}$	Adjacent matrix of B.
$\tilde{J} \in \mathbb{R}^{N \times N}$	The adjacent matrix plus self-connection.
$r_t \in \mathbb{R}^{N \times N}$	Vector of request rate of all nodes in graph A at time t.
$r^i_{t,average}$	The historical average content request rate at time epoch t of region $a_i \in A$.
$r^i_{t,median}$	The historical median content request rate at time epoch t of region $a_i \in A$.
$r^i_{t,min}$	The historical minimum content request rate at time epoch t of region $a_i \in A$.
$r^i_{t,max}$	The historical maximum content request rate at time epoch t of region $a_i \in A$.
d^i_t	The historical standard deviation at time epoch t of region $a_i \in A$.

According to the system assumptions, the catalog of IoT sensing data is defined as $F = \{1, ..., F\}$ carrying the Freshness Period (FP) parameter. The set of BBU pools $B = \{1, 2, ..., B\}$ equipped with limited storage for caching connects the Base Stations (BSs) which monitor the request data generated from the UEs. The set of IoT gateways $G = \{1, 2, ..., G\}$ with storage capabilities to store sensing data such that BBU pools can collect data directly from G through backhaul links instead of sensors. Avoiding activating sensors frequently helps to reduce energy consumption. Thus, we can attach a cache to GW or BBU pool to store sensing data which is not only popular but also fresh. The request rate of a certain data at the tth time slot (e.g., 10 min) of area $a_i \in A$ is defined as $r_t \in \mathbb{R}^N$ (N is the number of areas covered by the BSs). More specifically, we calculate the following parameters in the general time interval t:

- the sum of received sensing data requests, $N_{tot,t}$.
- the count of requests of every distinct requested data, $N_{dis,t}$.

The request rate of a certain data at interval t, $r_t^{v_i}$, is then obtained as $r_t^{v_i} = \frac{N_{dis,t}}{N_{tot,t}}$, which is also the ith element of r_t.

Data request rate forecasting refers to predicting future states of the user request data, given previously observed states of the user request data from an IoT network. IoT networks are time-vary dynamic. Therefore, as a basic decision parameter, data freshness also affects IoT data caching strategies. Data measurements such as road congestion and temperature will become stale after a short period. Thus, in our work, we also take into account the freshness of data. The IoT network with large-scale BSs distribution is transformed into a graph network containing N vertices that depict N areas covered by BSs and a set of edges that depict adjoining areas. As a classic time series forecasting problem, the most recent real-time measurement data can contribute valuable information for data request rate prediction. Moreover, historical statistics can also assist us to predict data request rates in the near future.

The daily trend of the request rate status of sensing data requests can be obtained by utilizing historical statistics as the input of the prediction model. During the t_{th} time slot of area $a_i \in A$, the historical average data request rate, median data request rate, minimum data request rate, maximum data request rate, and standard deviation re denoted by $r^i_{t,average}, r^i_{t,median}, r^i_{t,min}, r^i_{t,max}, d^i_t$, respectively. The task of data request rate prediction is to predict the future value of each region in a certain period via learning the previously obtained records. We summarize the main aforementioned notations in TABLE I for data request rate prediction.

Thus, the data request rate prediction problem can be formulated as:

$$r_{T+1} = Pr([r_1, r_2, \cdots, r_t, \cdots, r_T]; B(A, J)), \tag{4}$$

where $R_T = [r_1, r_2, \cdots, r_t, \cdots, r_T]$ represents historical graph information with T time steps. $Pr(\cdot)$ is a conditional probability function that attempts to predict the signal in the next step.

C. Request Graph Convolution LSTM

In this subsection, the novel Request Graph Convolutional Long Short-Term Memory (RGC-LSTM) integrating spatial-temporal knowledge and historical statistics into the deep learning model for data request rate prediction is proposed. In our model, we replace the inputs of the LSTM with the graph convolution features and ensure that other structures of LSTM remain unchanged. The graph convolution-based adjacency matrix has been defined in previous work [21], [14], [28]. The idea of a convolutional layer in the graph deep learning network is to learn the influence of the characteristics of each node in the graph structure data on its neighboring nodes. Generally, the input data is characterized by a 2D or 3D matrix structure. The propagation rules of each convolutional layer can be defined as the following expression:

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{J} \tilde{D}^{-\frac{1}{2}} H^l W^l), \qquad (5)$$

where $\tilde{J} = J + I$ is the sum of the adjacent matrix and self-connection of the undirected graph G, I is a matrix whose diagonals are all ones (i.e., the identity matrix). This allows the characteristic information of the vertex itself to be retained when the information is propagated. $\tilde{D}$ is the degree matrix of $\tilde{J}$ denote as: $\tilde{D} = \text{diag}(\sum_j \tilde{J}_{ij})$ and H^l is the activation unit matrix of layer L. The embedding matrix's initial value is the node feature: $H^0 = X$. W^l is the trainable matrix of each layer. σ is usually the activation function of ReLU, which is applied in all layers but the output layer. Different from the feedforward layer, the graph convolutional layer perform the multiplication with $\tilde{D}^{-\frac{1}{2}} \tilde{J} \tilde{D}^{-\frac{1}{2}}$ in the front. On the other hand, the multiplication with $\tilde{D}^{-\frac{1}{2}} \tilde{J} \tilde{D}^{-\frac{1}{2}}$ can also be interpreted as a collection of embeddings of adjacent nodes. The weight matrix W is the parameter of the different layers l of GCN.

GCN is a convolutional neural network that stacks up multiple layers of graphs. On each convolutional layer, GCN only processes the first-order adjacent message of a certain node, thus, the neighborhood matrix J of distributed BSs graph, the input data r_t can be substituted into Equation (5) to extract characteristics from the one-order neighborhood. Via superimposing several convolutional layers, multi-level neighborhood information transmission is realized, i.e., we can still retrieve convolutional filter capabilities via stacking multiple such layers. The frequently used two-layer GCN can be elegantly written as:

$$R_t^{(1)} = \text{RELU}(\hat{J} \cdot \text{RELU}(\hat{J} R_t^0) W^{(0)} \cdot W^{(1)}), \qquad (6)$$

where $\hat{J}$ is a standardization of J denoted as: $\hat{J} = \tilde{D}^{-\frac{1}{2}} \tilde{J} \tilde{D}^{-\frac{1}{2}}$.

More specifically, first we stack multiple graph convolution layer to capture the spatial dependencies. Subsequently, the spatial-temporal variable R_t^K is fused with historical variable H_t (including the historical average content request rate, median content request rate, minimum content request rate, maximum content request rate, and standard deviation at) to construct the input vector $P_t = [R_t^K; H_t]$, where $P_t \in \mathbb{R}^{NN}$, two tensors are concatenated by operator $[\cdot; \cdot]$ along the same dimensions. P_t is then fed into the encoder of LSTM model. At time epoch t forget gate, input gate, output gate, and input

cell state are represented by fg_t, ip_t, op_t, and $\tilde{S}_t$ respectively. The calculation formulas are defined as follows

$$fg_t = \sigma(W_f \cdot P_t + V_f \cdot h_{t-1} + bt_f), \qquad (7)$$

$$ip_t = \sigma(W_i \cdot P_t + V_i \cdot h_{t-1} + bt_i), \qquad (8)$$

$$op_t = \sigma(W_o \cdot P_t + V_o \cdot h_{t-1} + bt_o), \qquad (9)$$

$$\tilde{S}_t = \tanh(W_S \cdot P_t + V_S \cdot h_{t-1} + bt_S), \qquad (10)$$

$\cdot$ is the matrix multiplication operator. W_f, W_i, W_o and $W_S \in \mathbb{R}^{N \times NN}$, which map the input to the corresponding gate and cell are the weight matrices of LSTM. V_f, V_i, V_o and $V_S \in \mathbb{R}^{N \times N}$ are the weight matrices of hidden states. The bias terms are represented by bt_f, bt_i, bt_o, and $bt_S \in \mathbb{R}^N$, respectively. The σ is the activation function (typically tanh for status and output and sigmoid function for gates). We then can calculate the final cell state and the hidden state:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{S}_t, \qquad (11)$$

$$h_t = op_t \odot \tanh(C_t), \qquad (12)$$

At the ultimate time epoch T, the outcome of RGC-LSTM is the hidden state h_T, i.e., the predicted result $\hat{y}_t = h_T$. Let $y_T \in \mathbb{R}^N$ represent the label of the input data $P_T \in \mathbb{R}^{N \times N}$. For sequential data request prediction problem in our work, labels of time T ($T \in \{..., n-1, n, n+1, ..., n+m\}$) is the input of the immediate next time $(T+1)$, i.e., $y_T = p_{T+1}$. Thus, we define the loss function during training process as $L(y_T, \hat{y}_T) = L(p_{T+1}, h_T)$, i.e., the residual between the predicted result $\hat{y}_t$ and the labels is calculated by $L(\cdot)$, which is typically Mean Squared Error(MSE) function (MSE).

V. HIERARCHICAL PRE-CACHING STRATEGY

Average retrieve latency (ARL), acting as an important indicator to evaluate the performance of network caching, is the time interval between the user requesting data and the user receiving the data packet. We formulate the energy-saving problem in the complicated non-stationary temporal dynamical and spatial dependent green IoT networks as an optimization problem that decides to cache and update data at each time unit to minimize the ARL constrained by the storage capacity in this section. Accordingly, we design a heuristic algorithm to work out the above optimization problem.

A. Optimization Formulation of Interesting Data Pre-cacheing

Given the status of the distribution of user requests for content, once receiving a request for data item f, we first search in its associated BBU pool b and download it to the user if it is in b. If the search fails, the BBU pool b requests data from the IoT GW g through the backhaul. Then, if the requested data is in the cache and is up-to-date, the GW directly sends it to the BBU pool or otherwise collects it from corresponding sensors to relay it. Here we define R_{ab}^f as the request rate for f from area a to BBU b. D_{ab}, D_{bg} and D_{gs} represents the physical transmission latency between area a and BBU b, BBU b and GW g, GW g and sensor s, respectively. Θ_F, dynamically updated at each interval t

TABLE II
NOTATIONS FOR MINIMIZATION PROBLEM.

Notations	Description
F	Catalog of IoT contents $C = \{1, ..., F\}$.
FP	Freshness Period parameter carried by IoT contents.
B	Set of BBU pools $B = \{1, 2, ..., B\}$.
G	set of IoT gateways $G = \{1, 2, ..., G\}$.
c_i	Storage capacities of IoT network entity i
C_i	Upper limit budget of total caches of IoT network entities i.
R_{ab}^f	Request rate for f from area a to BBU b.
D_{ij}	Physical transmission latency between IoT network entity i and j.
X_i^f	Boolean variable to indicate whether file f is obtained from IoT network entity i.

is a freshness threshold that helps to identify whether the requested IoT content long-lived or short-lived. We denote the Exponential Weighted Moving Average (EWMA) of the average Freshness Period ($\overline{FP}$) as Θ_F:

$$\Theta_F(t) = (1 - \alpha)\Theta_F(t - 1) + \alpha\overline{FP}, \tag{13}$$

Since we have predicted the request rate for content f R_{ab}^f in area $a \in V$ at t^{th} time slot in Section IV. We can get the ARL function at time t:

$$ARL = \sum_{f \in F} \sum_{a \in V_b} 2R_{ab}^f(t)\{D_{ab}X_b^f(t) + (D_{ab} + D_{bg})$$
$$[1 - X_b^f(t)]X_g^f(t) + (D_{ab} + D_{bg} + D_{gs})$$
$$[1 - X_b^f(t)][1 - X_g^f(t)]\}, \tag{14}$$

Two Boolean variables X_b^f and X_g^f are defined to indicate whether sensing data f is obtained from BBU b and GW g, respectively. We list the main aforementioned notations in TABLE II for formulating the average retrieve latency minimization problem.

With the consideration of data freshness, the average retrieve latency minimization problem is defined as:

$$min \sum_{f \in F} \sum_{a \in V_b} 2R_{ab}^f(t)\{D_{ab}X_b^f(t) + (D_{ab} + D_{bg})$$
$$[1 - X_b^f(t)]X_g^f(t) + (D_{ab} + D_{bg} + D_{gs})$$
$$[1 - X_b^f(t)][1 - X_g^f(t)]\}, \tag{15}$$

$$s.t. X_b^f(t), X_g^f(t), Y_b^f(t), Y_g^f(t) \in \{0, 1\}, \ t \in \{0, 1, 2, ...\}, \tag{16}$$

$$X_b^f(t) = \begin{cases} 1, & if \ Y_b^f(t) = 1 \ and \ FP_f(t) > \Theta_F(t) \\ 0, & otherwise, \end{cases} \tag{17}$$
$$\forall f \in F, \ t \in \{0, 1, 2, ...\},$$

$$X_g^f(t) = \begin{cases} 1, & if \ Y_g^f(t) = 1 \ and \ FP_f(t) > \Theta_F(t) \\ 0, & otherwise, \end{cases} \tag{18}$$
$$\forall f \in F, \ t \in \{0, 1, 2, ...\},$$

$$\sum_{f=1}^{F} Y_b^f(t) \leq c_b, \ \forall b \in B, t \in \{0, 1, 2, ...\}, \tag{19}$$

$$\sum_{f=1}^{F} Y_g^f(t) \leq c_g, \ \forall g \in G, t \in \{0, 1, 2, ...\}, \tag{20}$$

$$\sum_{b=1}^{B} c_b \leq C_b, \ \forall b \in B\{1, 2, ..., B\}, \tag{21}$$

$$\sum_{g=1}^{G} c_g \leq C_g, \ \forall g \in G\{1, 2, ..., G\}. \tag{22}$$

The objective function is shown in Equation (15). In each period, Eq. (15) decides whether to store the data in the cache of BBU pool or the cache of IoT GW to minimize the average data retrieval latency. Eq. (16) denotes that $Y_b^f(t)$ and $Y_g^f(t)$ are also binary variables which indicate whether data $f \in F\{1, 2, ..., F\}$, are stored in the BBU b and GW g, respectively. Equations (17) and (18) imply the relationship of $X_b^f(t)$, $Y_b^f(t)$, $\Theta_F(t)$, and the relationship of $X_g^f(t)$, $Y_g^f(t)$, $\Theta_F(t)$, respectively. Once the user request is received, the requested sensing data is retrieved from the BBU pool first, sensing data f is obtained from the BBU b (i.e., $X_b^f = 1$) if 1) sensing data f is cached in the BBU (i.e., $Y_b^f = 1$) and 2) sensing data f is fresh (i.e., $FP_f(t) > \Theta_F(t)$). Otherwise, sensing data request is forwarded to IoT GW (i.e., $X_b^f = 0$) if 1) sensing data f is cached in the BBU but is stale (i.e., $Y_b^f = 1$ and $FP_f(t) < \Theta_F(t)$) or 2) sensing data f does not exist in the BBU b (i.e., $Y_b^f = 0$). Data f is obtained from the GW g (i.e., $X_g^f = 1$) if 1) sensing data f is in the GW (i.e., $Y_g^f = 1$) and 2) sensing data f is fresh (i.e., $FP_f(t) > \Theta_F(t)$). Otherwise, sensing data f is collected from sensors (i.e., $X_g^f = 0$) if 1) sensing data f is in the GW but is stale (i.e., $Y_g^f = 1$ and $FP_f(t) < \Theta_F(t)$) or 2) data f does not exist in the GW g (i.e., $Y_g^f = 0$). Eqs. (19) and (20) indicate that the sum of whole data sizes per BBU and GW should not exceed their storage capacity, respectively. Eqs. (21) and (22) indicate that the cache attached to all BBU pools and GWs should not exceed the storage thresholds C_b and C_g, otherwise it will consume too many resources.

Although it is possible to predict the complete information of the network state within a specific period in practice, it is still challenging to solve the problem due to its high computational complexity.

B. RGC-LSTM-Based Pre-Caching Greedy Algorithm

In this subsection, considering time efficiency, a heuristic algorithm, i.e., RGC-LSTM-Based hierarchical pre-caching algorithm is proposed to obtain a fast solution.

The cache placement policy determines data to be stored at each BBU pool and IoT GW to decrease the ARL. We can draw the following conclusions via combining formula analysis and experience:

- Intuition 1: Caching data with higher request rate can reduce more latency.
  We express ARL as Eq. (11). If we have $X_b^1(t) = X_g^1(t) = X_b^2(t) = X_g^2(t) = 0$ and $R_{ab}^1 > R_{ab}^2$, cache 1 has a lower delay than 2 due to the more reduction of the objective function.
- Intuition 2: Along the path from sensors to the UEs, caching closer to UEs reduces more latency.

Algorithm 1 RGC-LSTM-Based Pre-Caching Greedy Algorithm.

Input: $B, G, c_b, c_g, C_b, C_g, R_{ab}^f, F, FP, D_{ab}, D_{bg}, D_{gs}, X, Y$
Output: Average retrieve latency ARL, content placement solutions Y_b, Y_g at all time epochs;

1: Initialization of time epoch t = 0;
2: **for** each time epoch t **do**
3:      **for** each BBU b **do**
4:          Load the total request rate of data f of all BSs related to the BBU from the prediction result of the RGC-LSTM.
5:          **for** each predicted data f in descending sort of R_f **do**
6:              **if** $FP_f(t) > \Theta_F(t)$ **then**
7:                  Cache data f until BBU b is insufficient;
8:              **else**
9:                  Request data f with IoT gateway through backhaul links;
10:                  Update FP_f;
11:      **for** each GW g **do**
12:          **for** each predicted data f from all region in descending order of request rate **do**
13:              **if** data f is not in BBU pool b and $FP_f(t) > \Theta_F(t)$ **then**
14:                  Cache data f until GW g is insufficient;
15:              **else**
16:                  Activate the related IoT sensors to transmit the data f;
17:                  Update FP_f;
18:      All Freshness Period parameters in FP automatically subtracted by 1.
19:      $t \leftarrow t + 1$;
20:      Calculating new freshness threshold: $\Theta_F(t) = (1 - \alpha)\Theta_F(t-1) + \alpha\overline{FP}$;

We consider two nodes $b \in B$ and $g \in G$. For data f, when pre-caching at node b but not at node g, the total latency will decrease by D_{bg}, so caching closer to UEs will reduce more retrieving latency.

Based on intuitions 1 and 2, we conclude that hierarchically data caching should be realized in the network. Frequently requested and fresh data should be closer to the UEs, i.e., storing data with a higher request rate as well as larger FP demonstrates high-performance when designing caching strategies. Greedy idea is utilized to store as much data with a higher request rate as possible at each network entity. More specifically, under the condition of $FP_f(t) > \Theta_F(t)$, our State-awareness-based greedy algorithm first stores the data with the highest request rate at each BBU pool b until reaching its maximum storage capacity. Then, each GW stores the data with the highest request rate until reaching its maximum storage capacity. Algorithm 1 details the energy-efficient pre-caching strategy. Lines 4–11 describe the caching process per BBU pool. Caching in each GW is depicted between lines 12–20. Based on the prediction results of RGC-LSTM and in a specific time interval, the time complexity of the second

for loop (i.e., between lines 2 and 10) is $O(BF)$. The worst-case time complexity of the third for loop (i.e., between lines 11 and 17) is $O(GF)$. Combining the two, the computational complexity of our proposed algorithm is $O(BF + GF)$. Our proposed algorithm which allows leveraging the cache to temporarily store sensing data requested by the user (i.e., cache in GW and BBU pools), formulates reasonable storage allocation and data placement in each time period. GW and BBU pools can directly provide IoT data to users without collecting them from sensors after receiving user requests, which can avoid activating sensors frequently(consuming considerable energy).

VI. EVALUATION

The evaluation goes here. Compared with the traditional Popular-based algorithm, we consider the performance of our proposed heuristic algorithm. Experiment results show that our proposed heuristic algorithm achieves substantial energy reduction.

A. Experiment Settings

In our simulations, we assign the number of BBU pools B = 4 and the number of IoT gateways G = 2, and the cache capacity is C_b =1000 MB and C_g = 3000 MB, respectively. For simplicity, we specify the sizes of IoT data are 1 MB. Moreover, the freshness period of the IoT data triggering the discarding of expired data is randomly chosen from 2 to 4 time units.To simulate data request distribution in the real world, Zipf distribution model [29] is adopted, i.e., the distribution of sensing data requests from an area covered by BS conforms to Zipf's law. We employ NDN Traffic Generator [30] to generate user requests for IoT data in each region, where the frequency of data accords to Zipf-like distributions. According to the base station distribution map in the city of Ghent [25], We constructed a joint spatio-temporal dataset, which collects the measurement parameters of data requests in all 13 regions. Due to the huge difference in the range of measured parameter values, hence the dataset requires to be normalized before model training.

RGC-LSTM is utilized for multi-step prediction of time-series data (i.e., input is multiple sequences, output is one sequence). In this paper, we calculate the request for certain data in each region as the average value within 5 minutes, thus the clock time is converted into an ordered integer N, e.g., $00:00 - 00:05$ as $N_t = 1$ and $12:00 - 12:05$ as $N_t = 145(12 * 12 + 1)$. We extracted 60 days of data generated by the traffic generator for experimental purposes. The extracted data set is divided into a training set comprised of records from the previous 50 days, and the testing dataset consisting of the remaining observations in the next 10 days. The prediction time horizon is set as 00:00-24:00, thus, for certain data, each region contains 288 data points per day. The simulations are implemented by Python with the Pytorch platform. We choose Adam optimizer (learning rate 0.01) and train our prediction model via minimizing loss function (MSE).

B. Evaluation Metrics

In the part of perceiving requests for interesting data, inputs are continuous time-series data with 4 time units. The performance of our proposed RGC-LSTM is evaluated by the commonly used metric for predicting Mean Squared Error (MSE) function: $MSE = \frac{1}{n} \sum_{n=1}^{N} (y_T - \hat{y}_T)^2$.

In the part of verifying the performance of the cache strategy, our proposed autonomous pre-caching scheme is compared against the Popular-based scheme. Popular-based always uses the idea of greed to cache the most popular data in network entities at each time period. On the other hand, RGC-LSTM learns the characteristics of localized data requests, which help train a good predictive model. Based on the prediction result, RGC-LSTM can always cache the freshest and most needed data in the network entity near the user. We consider the following performance metrics.

- Cache Hit Ratio: Cache Hit Ratio (CHR) is the mean ratio of the number of responses to the sum of requests from users within a specific time. Values are expressed in percentage as

$$CHR = \frac{N_{resp}}{N_{count}}, \qquad (23)$$

where N_{resp} represents the sum of response number from network entities, N_{count} represents the sum of requests received by network entities. The value of CHR reflects the pros and cons of the cache strategy. A high CHR value indicates the effectiveness of the caching scheme and low data redundancy.

- Average Retrieve Latency: Average Retrieve latency(ARL) is calculated as the average time for a UE to retrieve the IoT data.

- Average Sensor Activation Rate(ASAR): Power consumption is a traditional measure of the energy consumption of the sensor. It can be obtained by monitoring the power consumption of the sensor in real-time. However, we can indirectly measure its energy consumption by counting the number of activations of the sensor, because once the data is not cached in the IoT network, the GW will activate the relevant sensors to transmit interesting data. Thus, we define the Average Sensor Activation Rate as follows:

$$ASAR = \frac{N_{outdate} + N_{notin}}{N_{total}}, \qquad (24)$$

where N_{total} is the total number of interesting data requested in all regions in each time unit, and $N_{outdate}$ and N_{notin} are the number of data that is stale and not cached in the IoT network, respectively. The greater the ASAR, the greater the energy consumption.

C. Experiment Results

Fig.2(a) shows a comparison histogram that when the graph convolution aggregates the information of the surrounding neighborhood nodes in RGC-LSTM, the impact of the selection of the number of hops on the training time per epoch. The result shows that the training time per epoch increases with the increase in the value of K. In our simulation, the training

time in each epoch increases significantly when K exceeds 3. Longer training time will result in more computational and energy consumption. Fig. 2(b) also illustrates the impact of the selection of the number of hops on the training losses when the graph convolution aggregates the information of the surrounding neighborhood nodes in RGC-LSTM. As illustrated in Fig. 2(b), the increase in the value of K speeds up the convergence. However, for our datasets, the training outcome improves only indistinctively when K is larger than 3. The choice of the number of hops in the graph convolution may be different due to the model evaluation and comparison. Note that $K = 3$ has advantages in both training time and accuracy. Thus for designing future a pre-caching strategy, we set it as 3, which simultaneously accuracy and performance.
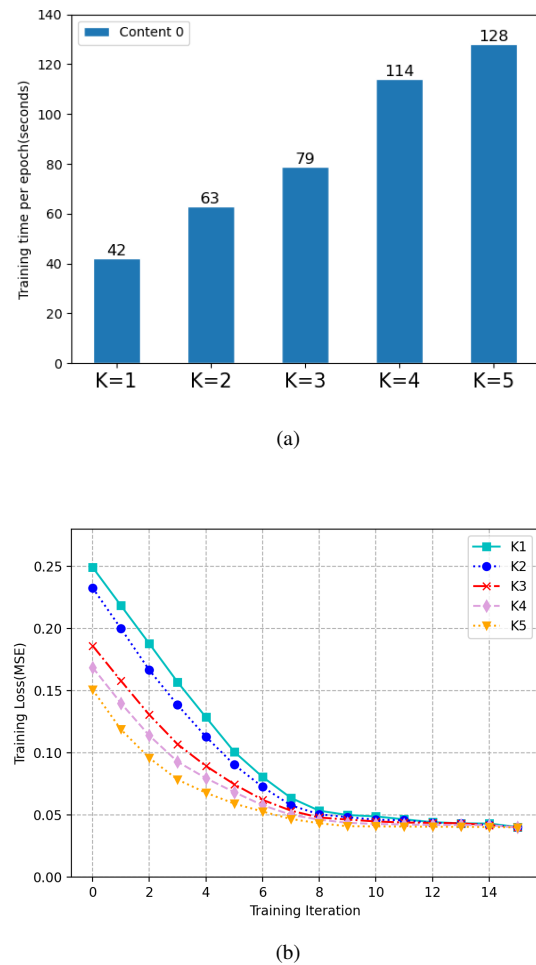

(a)

(b)

Fig. 2. Influence of neighboring orders (hops) of the graph convolution. (a) Training time per epoch. (b) Training losses.

Fig. 3 visualizes the prediction values and the ground truth of the request for certain content in a region. As shown in Fig.3, two curves fit well, indicating that after the graph convolution operation (i.e., aggregating the information of the neighborhood), our model can predict the trend of requests well in different time epoch.

In the part of comparing cache strategies, we analyze the performances of our algorithm with Popular-based algorithms
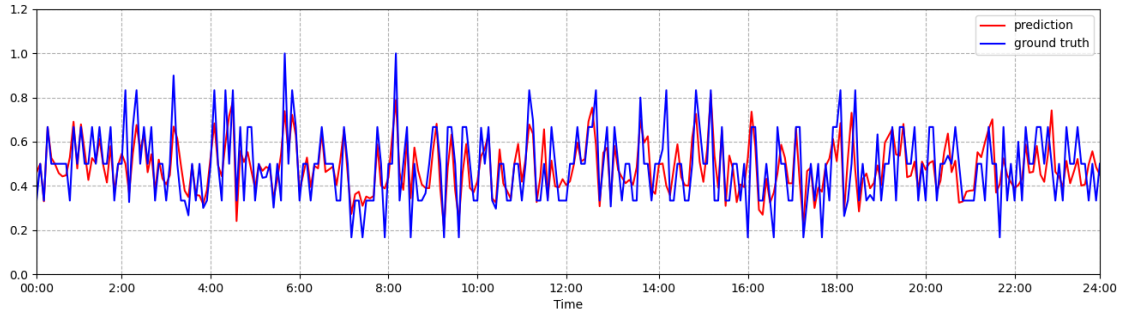
Fig. 3. Visualization of prediction and ground truth of request for content 0 on a randomly selected day.
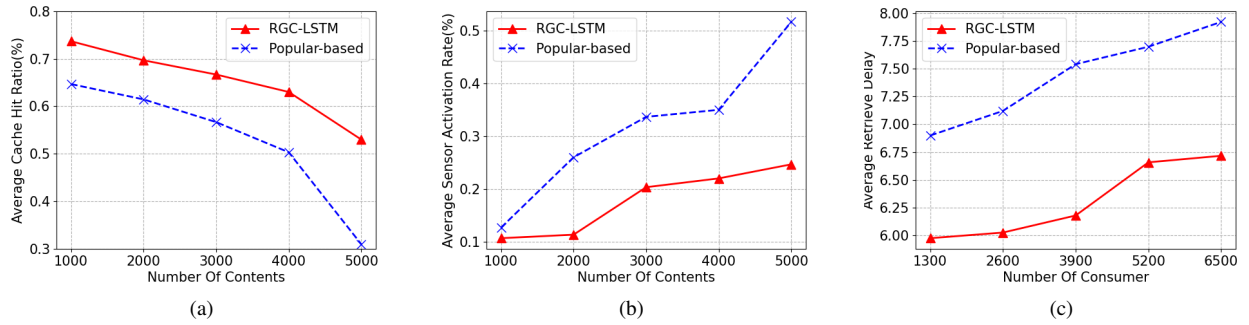

Fig. 4. Impact of the number of content and consumers in green IoT on key evaluation indicators. (a) Average cache hit ratio. (b) Average sensor activation rate. (c) Average retrieve delay.

from three different evaluation metrics, i.e., the average cache hit ratio in Fig. 4(a), the average retrieve delay in Fig. 4(b), and the average sensor activation rate in Fig. 4(c). Note that RGC-LSTM works better than Popular-based algorithms, i.e., the highest average cache hit ratio, the minimum average data retrieve delay, and the minimum average sensor activation rate. Moreover, in Fig. 4(a), we first compare the proposed algorithms with Popular-based algorithms under the same condition (e.g., cache capacity C_b =1000 MB and C_g = 3000 MB). As illustrated in Fig. 4(a), the larger the number of contents, the lower the average cache hit ratio. This is because the network entity has a limited storage space and is capable to individually hold a fixed amount of data. Compared with the other scheme, the CHR of RGC-LSTM can reach 0.53 even the number of contents is 5000. Moreover, the CHR of Popular-based is 0.31. Popular-based does not consider the relationships between data requests in different regions and different time periods, but only uses the idea of greed to design a caching strategy in each time period. Fig. 4(b) also depicts the result by setting the number of contents to range from 1000 to 5000. It resulted in fewer sensor activation than the case of Popular-based caching. As shown in Fig. 4(b), a huge amount of data make for a larger number of sensor activations for both algorithms while the increase rate in the number of sensor activations of RGC-LSTM is significantly lower than that of Popular-based algorithms. This is because although network entities have limited storage capacity, which allows them to store only a certain amount of data, RGC-LSTM can accurately predict data requests at each moment,

which makes it more reasonable to allocate cache locations and content. We learn the influence of the number of consumers with the same condition. Fig. 4(c) illustrates the relationship between key evaluation metrics and the number of consumers. As illustrated in Fig. 4(c), it indicates that once the number of consumers increases, the average retrieve delay becomes longer for both algorithms because the increase of consumers requesting interesting data injects massive traffic into IoT networks, resulting in network congestion. The advantages of low increase of our RGC-LSTM get more and more clear, with the increase of consumer requests. Therefore, the proposed RGC-LSTM algorithms can be considered as a green pre-caching algorithm, which is more applicable in blockchain-based green IoT.

D. Discussion

In this work, we deploy RGC-LSTM and make pre-caching policies in edge-AIs, which are designed to process massive information streams and deploy AI algorithms (adaptive boost-inting, deep learning, etc). We concentrate on securing the edge-AIs from malicious attacks and while minimizing energy consumption. The edge-AIs share their data via blocks in a private blockchain. All participating parties have high transparency and credibility. Once an edge-AI is compromised, it is relatively simple to withdraw the previous set of operations on a private blockchain. Moreover, the transparent, tamper-proof ledger in private blockchain stores detailed information related to the transaction, such as shard location, shard hash, etc.

VII. CONCLUSION

In this paper, to minimize sensors' energy consumption while providing quality of service (QoS) in IoT networks, we proposed a sophisticated deep learning architecture for reducing energy consumption. The GCN and LSTM are united to learn the spatial-temporal dependencies in the green IoT network for request-state prediction. In addition, based on the prediction, we have proposed a heuristic algorithm implementing the hierarchical pre-caching process, where contents can be cached at both the BBU and GW with much lower time complexity. By evaluating on spatial-temporal joint distribution datasets, our strategy is demonstrated to be superior compared to the conventional Popular-based methods. Moreover, we utilized blockchain technology in the proposed architecture to build P2P communication for strict access to edge-AIs and secure data sharing, preventing an internal node from deliberately concealing or tampering with data and thus achieving complete privacy. Future studies could include improving the accuracy and robustness of our model and performing experiments on large urban networks.

REFERENCES

[1] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, and I. Godor, "Ericsson mobility report," *Pulse of the Networked Society*, 2015.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] J. Chen, J. Wu, H. Liang, S. Mumtaz, J. Li, K. Konstantin, A. K. Bashir and R. Nawaz, "Collaborative trust blockchain based unbiased control transfer mechanism for industrial automation," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4478-4488, 2019.

[4] H. Liang, J. Wu, S. Mumtaz, J. Li, X. Lin and M. Wen, "MBID: Micro-blockchain based geographical dynamic intrusion detection for V2X," *IEEE Communications Magazine*, vol. 57, no. 10, pp. 77-83, 2019.

[5] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," arXiv:1809.00343, 2018.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash,"Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] X. Sun and N. Ansari, "Traffic load balancing among brokers at the IoT application layer," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 489-502, 2017.

[8] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong and R. Tafazolli, "In-network caching of Internet-of-Things data," *IEEE International Conference on Communications*, DOI:doi: 10.1109/ICC.2014.6883811, 2014.

[9] J. Yao, N. Ansari, "Caching in dynamic IoT networks by deep reinforcement learning," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2020.3004394, 2020.

[10] H. Wang, Y. Li, X. Zhao, and F. Yang, "An algorithm based on markov chain to improve edge cache hit ratio for blockchain-enabled IoT," *China Communications*, vol. 17, no. 9, pp. 66-76, 2020.

[11] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[12] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, "Spectral networks and deep locally connected networks on graphs," arXiv:1312.6203, 2013.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, arXiv:1606.09375, 2016.

[14] Z. Zhou and X. Li, "Convolution on graph: A high-order and adaptive approach," arXiv:1706.09916, 2018.

[15] F. A. Gers, J. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, pp. 2451–2471, 2000.

[16] X. Sun and N. Ansari, "Dynamic resource caching in the IoT application layer for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 606–613, 2018.

[17] J. Quevedo, D. Corujo, and R. Aguiar, "A case for ICN usage in IoT environments," *IEEE Global Communications Conference*, DOI: 10.1109/GLOCOM.2014.7037227, 2014.

[18] J. Wu, M. Dong, K. Ota, J. Li, W. Yang, "Application-aware consensus management for software-defined intelligent blockchain in IoT," *IEEE Network*, vol. 34, no. 1, pp. 69-75, 2020.

[19] P. Duan et al., "Space-reserved cooperative caching in 5G heterogeneous networks for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2715–2724, 2018.

[20] J. Wu, Z. Zhang, Y. Hong and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35-41, 2015.

[21] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," arXiv:1609.02907, 2017.

[22] M. Simonovsky and N. Komodakis, "Dynamic edge conditioned filters in convolutional neural networks on graphs," *IEEE Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/CVPR.2017.11, 2017.

[23] A. Narayanan, S. Verma, E. Ramadan, P. Babaie, and Z. L. Zhang, "Deepcache: A deep learning based framework for content caching," *Proceedings of the 2018 Workshop on Network Meets AI & ML*, DOI:10.1145/3229543.3229555, 2018.

[24] H. Mou, Y. Liu and L. Wang, "LSTM for mobility based content popularity prediction in wireless caching networks," *IEEE Globecom Workshops (GC Wkshps)*, DOI: 10.1109/GCWkshps45667.2019.9024419, 2019.

[25] S. K. Goudos, M. Deruyck, D. Plets, L. Martens, and W. Joseph, "Optimization of power consumption in wireless access networks using Differential Evolution with eigenvector based crossover operator," *European Conference on Antennas and Propagation*, DOI: 10.1109/EuCAP.2016.7481359, 2016.

[26] R. Ying, R. He, K. Chen, P. Eksombatchai, L. H. William, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, DOI: 10.1145/3219819.3219890, 2018.

[27] D. K. Hammond , P. Vandergheynst, and R. Gribonva, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129-150, 2011.

[28] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," arXiv:1506.05163, 2015.

[29] A. Tatar, M. D. De Amorim, S. Fdida, and P. Antoniadis, "A survey on predicting the popularity of Web content," *Journal of Internet Services and Applications* , vol. 5, no. 1, pp. 1-20, 2014.

[30] Accessed: May 2017. [Online]. Available: http://named-data.net/

Ge Chen received the B.S. degrees in network engineering from North China Electric Power University, Baoding, China, in 2019. Now, he is a candidate M.S. at School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include blockchain, graph neural networks.

Jun Wu received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011. He was a Post-Doctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He was a Researcher with the Global Information and Telecommunication Institute, Waseda University, Japan, from 2011 to 2013. He is currently a professor of School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He is also the vice dean of Institute of Cyber Science and Technology and vice director of National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiao Tong University, China. He is the chair of IEEE P21451-1-5 Standard Working Group. He has hosted and participated in a lot of research projects including National Natural Science Foundation of China (NFSC), National 863 Plan and 973 Plan of China, Japan Society of the Promotion of Science Projects (JSPS), etc. His research interests include the advanced computing, communications and security techniques of software-defined networks (SDN), information-centric networks (ICN), Internet of Things (IoT), 5G/6G, molecular communication, etc., where he has published more than 150 refereed papers. He has been the Track Chair of VTC 2019, VTC 2020 and the TPC Member of more than ten international conferences including ICC, GLOBECOM, etc. He has been a Guest Editor of IEEE Transactions on Industrial Informatics, IEEE Sensors Journal, Sensors. He is an Associate Editor of the IEEE Access, IEEE Networking Letters.
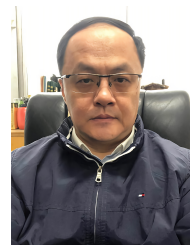
Wu Yang received the Ph.D. degree in computer system architecture specialty from the Computer Science and Technology School, Harbin Institute of Technology. He is currently a Professor and a Doctoral Supervisor with Harbin Engineering University. His main research interests include wireless sensor network, peer-to-peer network, and information security. He is a member of ACM and a Senior Member of CCF.

Ali Kashif Bashir received his Ph.D. in computer science and engineering from Korea University (83 in QS Ranking 2020) South Korea and MS from Ajou University, South Korea and BS from UMT, Pakistan. He is a Senior Lecturer at the Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom. He is also with School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad (NUST), Islamabad, Pakistan. He is a senior member of IEEE and Distinguished Speaker of ACM. His past assignments include Associate Professor of Information and Communication Technologies, Faculty of Science and Technology, University of the Faroe Islands, Denmark; Osaka University, Japan (71 in QS Ranking 2020); Nara National College of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. He is also advising several startups in the field of STEM based education, robotics, and smart homes. Currently, he is supervising RNS Solutions, a multinational organization as Chief Research Officer. He is author of over 80 peer-reviewed articles. He is supervising/co-supervising several graduate (MS and PhD) students. His research interests include internet of things, wireless networks, distributed systems, network/cyber security, cloud/network function virtualization, etc. He is serving as the Editor-in-chief of the IEEE FUTURE DIRECTIONS NEWSLETTER.

Gaolei Li (S'16, M'20) received the B.S. degree in electronic information engineering from Sichuan University, Chengdu, China, and Ph.D. degree in Cyber Security from Shanghai Jiao Tong University, Shanghai, China. From Oct. 2018 to Sep. 2019, he visited the Muroran Institution of Technology, Muroran, Japan, supported by the China Scholarship Council Program. Now, he is an Assistant Professor in School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His main research interests include cyber security, machine learning, and privacy protection. He has received best paper awards from the IEEE IEEE ComSoc CSIM Committee and student travel grant award for IEEE Globecom. He is the reviewer of IEEE TDSC, TII, IEEE Access, and TPC member for IEEE iThings 2016-2019, IEEE ICC 2018, IEEE ISPA 2020, ScalCom 2020&2021.

Mohammad Hammoudeh is a Senior Lecturer in Computer Networks and Security in the School of Computing, Mathematics and Digital Technology at the Manchester Metropolitan University. He received his Ph.D. in Computer Science from the University of Wolverhampton in 2009, MSc in Advanced Distributed Systems from the University of Leicester in 2007 and BSc (Hons) in Computer Communications from the Arts, Sciences & Technology University in Lebanon in 2004. He is the co-founder and member of the FUture Networks and Distributed Systems research Group (FUNDS). He is the founder and head of the MMU IoT Lab.