# Cognitive Popularity Based AI Service Sharing for Software-Defined Information-Centric Networks

**Article** *in* IEEE Transactions on Network Science and Engineering · April 2020

DOI: 10.1109/TNSE.2020.2993457



**7 authors**, including:

Ali Kashif Bashir
Manchester Metropolitan University
**263** PUBLICATIONS **3,833** CITATIONS
SEE PROFILE

Shahid Mumtaz
Institute of Telecommunications
**256** PUBLICATIONS **5,308** CITATIONS
SEE PROFILE

Alireza Jolfaei
Flinders University
**165** PUBLICATIONS **2,337** CITATIONS
SEE PROFILE

Nida Kvedaraitė
Kaunas University of Technology
**11** PUBLICATIONS **69** CITATIONS
SEE PROFILE

# Cognitive Popularity based AI Service Sharing for Software-Defined Information-Centric Networks

Siyi Liao, Jun Wu, Jianhua Li, Ali Kashif Bashir, Shahid Mumtaz, Alireza Jolfaei, and Nida Kvedaraite

***Abstract*—As an important architecture of next-generation network, Software-Defined Information-Centric Networking (SD-ICN) enables flexible and fast content sharing in beyond the fifth-generation (B5G). The clear advantages of SD-ICN in fast and efficient content distribution and flexible control make it a perfect platform for solving the rapid sharing and cognitive caching of AI services, including data samples sharing and pre-trained models transferring. With the explosive growth of decentralized artificial intelligence (AI) services, the training and sharing efficiency of edge AI is affected. Various applications usually request the same AI samples and training models, but the efficient and cognitive sharing of AI services remain unsolved. To address these issues, we propose a cognitive popularity-based AI service distribution architecture based on SD-ICN. First, an SD-ICN enabled edge training scheme is proposed to generate accurate AI service models over decentralized big data samples. Second, Pure Birth Process (PBP) and error correction-based AI service caching and distribution schemes are proposed, which provides user request-oriented cognitive popularity model for caching and distribution optimization. Simulation results indicate the superiority of the proposed architecture, and the proposed cognitive SD-ICN scheme has 62.11% improved to the conventional methods.**

***Index Terms*—cognitive popularity, decentralized big data, Software Defined Information-Centric Network (SD-ICN), service sharing.**

## I. Introduction

RECENTLY, Software-Defined Networking (SDN) and Information-Centric Networking (ICN) have been extensively studied as the mainstream architectures of the next-generation network. When SDN meets ICN, they will greatly enhance network management, such as traffic engineering, routing and service chaining [**?**]. The separation of control and forwarding of SDN and the well-developed OpenFlow protocol can be combined with the characteristics of dynamic naming and efficient content distribution of ICN. Therefore, as an integration of them, Software-Defined Information-Centric Networking (SD-ICN) has become an important content sharing and distributing network architecture in the beyond fifth-generation (B5G) [**?**]. It offers centralized control and in-network caching which makes it more ideal for a wide range of network devices. In the in-network caching and content distribution scheme of SD-ICN, every intermediate SD-ICN switch can provide interest or data on behalf of the original producer, reducing the so-called flash crowd situation [**?**]. Recent studies pointed out that content sharing service scales better in an SD-ICN architecture than the traditional host-centric IP model [**?**]. The clear advantages of SD-ICN in fast and efficient data transmission, content distribution and reliability assurance make it a very promising network model for AI service sharing, including data samples sharing and pre-trained models transferring [**?**].

Recognizing, discovering and extracting potential patterns from massive data is the core utility of big data analytics as it results in higher levels of insights for decision making and trend prediction [**?**]. Therefore, massive data capturing devices and sensors are deployed and distributed to gather continuous data for edge learning applications. Thanks to the recent advancement in fast computing, efficient storage and novel machine learning algorithms, more attention has been drawn in the area of big data analytics and knowledge extraction for diverse applications. Fast and efficient connectivity of these smart devices enables many valuable and remarkable applications like smart home, intelligent transport, e-health, smart grid, and smart cities [**?**]. Decentralized big data and machine learning tasks are fully integrated, bringing intelligence and cognition to the network.

However, the widespread deployment of edge learning also brings new problems that are not met by existing models (ie. homogenous learning models are repeatedly trained, models with smaller data volumes are easily over-fitting, etc.). The users will inevitably produce similar machine learning tasks, they need the same type of data, and even expect the same training results. For example, with the development of transfer learning, some models must be further trained on the existing initial model. This will also place higher demands on the sharing and distribution of the AI model. Without an effective sharing and fast distribution scheme, a large number of tasks will be repeatedly trained at the IoT edge and the same type of data cannot be aggregated and used as well. Whats more, a smaller amount of data samples will also lead to over-fitting of the model. Therefore, invalid and meaningless model training will be widespread, resulting in a waste of resources and a decline in Quality of Service (QoS). How to integrate decentralized data of the network for the training of AI models

This work was supported in part by the National Natural Science Foundation of China under Grant 61972255.(*Corresponding author: Jun Wu.*)

Jianan Li, Jun Wu and Siyi Liao are with Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, School of Cyber Security, Shanghai Jiaotong University, Shanghai 200240, China.(e-mail: junwuhn@sjtu.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, Islamabad (NUST), Islamabad, Pakistan.

Shahid Mumtaz is with Instituto de Telecomunicaes (IT), Portugal.

Alireza Jolfaei is with Department of Computing, Macquarie University, Sydney NSW 2109, Australia.

Nida Kvedaraite is with Kaunas University of Technology (KTU), Republic of Lithuania.

remains to be resolved. Therefore, both the decentralized big data resources and the machine learning models are in need of effective management and cognitive sharing scheme.

To address these issues, We make full use of the advantages of SD-ICN to enable the network devices with the ability to cognize the popularity of the AI service. The contributions of our work are summarized as follows.

- An SD-ICN based architecture that enables the efficient sharing and fast distribution of AI service is proposed, including data samples sharing and pre-trained models transferring. We have hierarchically defined the functions and relationships between the various functional models. Different packets in the proposed architecture are illustrated in detail.
- A novel scheme for effective edge training with decentralized big data is proposed. In order to solve the problem of repeat training and decline in model accuracy, similar AI services with the same type of samples and training results are aggregated and trained in the proposed SD-ICN architecture by using the decentralized big data.
- We propose a cognitive popularity model for the optimization of SD-ICN caching and distributing. We present a detailed mathematical model to optimize the cache space of the SD-ICN nodes and increase cache hit ratio based on Pure Birth Process (PBP) and error correction. Through the prediction and ranking of the request, we implemented a dynamic update of the cache.

The remainder of this paper is organized as follows. The related work is given In section II and the strengths of the proposed scheme are described. Our system model of scheme is presented in Section III. Both the basic implementation of the architecture and design principles analysis in detail are provided in Section IV. Simulation results are shown in Section V to estimate the performance of the scheme. Final conclusions are drawn in Section VI.

## II. Related Work

Related methods, including blockchain, AI and have been extensively studied so as to provide the network with faster and safer services [**?**] [**?**]. The development and expansion of IoT makes it one of the major sources of big data, as it connect a myrid of sensors and smart devices together to share their captured status of the environments [**?**]. This also enables the huge potential of edge AI and its related technologies, including applications in transportation, medical, and security [**?**] [**?**]. Voluminous amounts of data have been produced and used, since the past decade as the miniaturization and universalization of IoT devices [**?**]. Authors of [**?**] proposed a heuristic approach in the edge-cloud-hybrid system for IoT so as to increase the efficiency of big data service deployment. By the monitoring of multiple factors, an innovative system is presented for the detection and support of Obtrusive Sleep Apnea (OSA) of elderly people using the available open data [**?**]. A novel price forecasting model for Smart Grid is introduced in [**?**] by the integration of Differential Evolution (DE) and Support Vector Machine (SVM) classifier. Authors of [**?**] presented and discussed a scalable and flexible Deep Learning (DL) framework based on Apache Spark for mobile big data, which enables the orchestration of DL models with a large number of hidden layers and parameters on a computing cluster.

On the other hand, the architecture of the next-generation network has been widely discussed [**?**]. Based on the Software Defined Network (SDN) technology, authors of [**?**] proposed a content popularity prediction based on deep learning to achieve the popularity prediction. It uses the computing resource and link of SDN to build a distributed and reconfigurable deep learning network. As an content-centric approach, ICN have been recently regarded as an alternative to the traditional host-centric network paradigm [**?**]. Obvious benefits of ICN in terms of improved interest/content sharing scheme and better reliability has already raised ICN as highly promising networking techeology for environments such as IoT [**?**]. A novel cognitive ocean network (CONet) architecture is proposed as well as its important and useful demonstration applications [**?**]. Authors of [**?**] proposed an ICN-IoT architecture in which ICN nodes provide IoT gateways capabilities and ICN in-network caching. In order to improve the energy efficiency of IoT, the in-network caching of ICN is leveraged by authors of [**?**] to propose a novel cooperative caching scheme based on the IoT data lifetime and user request rate. Based on NDN, MR-IoT defines schemes to execute MapReduce tasks on IoT including computational tree construction and computational task dissemination [**?**]. With IoT and ICN combined all together with the Edge Computing concept, the cability of merging DL models is discussed and studied in [**?**], such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Reinforcement Learning (RL). Although the research on ICN and IoT has been extensive, few works have focus on applying ICN to the sharing of AI services and data locations in IoT. Efficient decentralized big data-based AI services are in urgent need.

## III. Basic Architecture

### A. Proposed Artificial Intelligence Service Sharing

AI-capable devices are widely deployed at the edge of the network and are deeply integrated with smart cities. The aggregation of massive decentralized data and the richness of edge computing resources at the edge of the network have stimulated a wide variety of artificial intelligence-based services and applications. However, in heterogeneous IoT networks, computing resources and edge data are often unbalanced, and it is almost impossible to train models based on every demand of user service. On the other hand, the extensive similar service has been repeatedly requested by various applications and users.

As shown in Fig.**??**, we regard the fog server as the basic unit of content caching and distribution, and form an information-centric network that connects massive IoT devices and enable the sharing of various AI-based services. In the proposed scheme, pre-trained Machine Learning (ML) models and decentralized big data are cached in the fog node. When a user requests a certain type of AI service, it first queries whether the service model has been cached in the local fog node. If

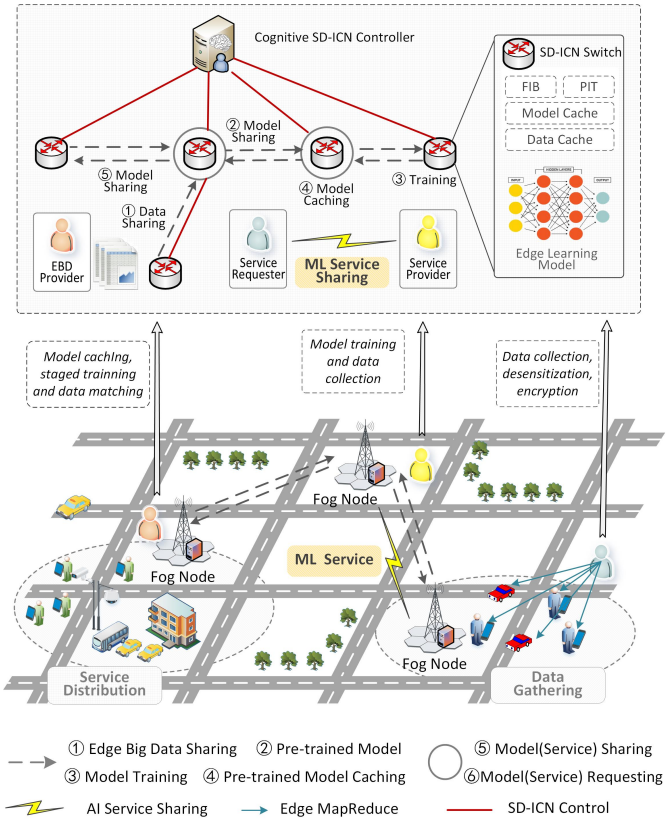


Fig. 1: Scenarios of AI service sharing using SD-ICN.

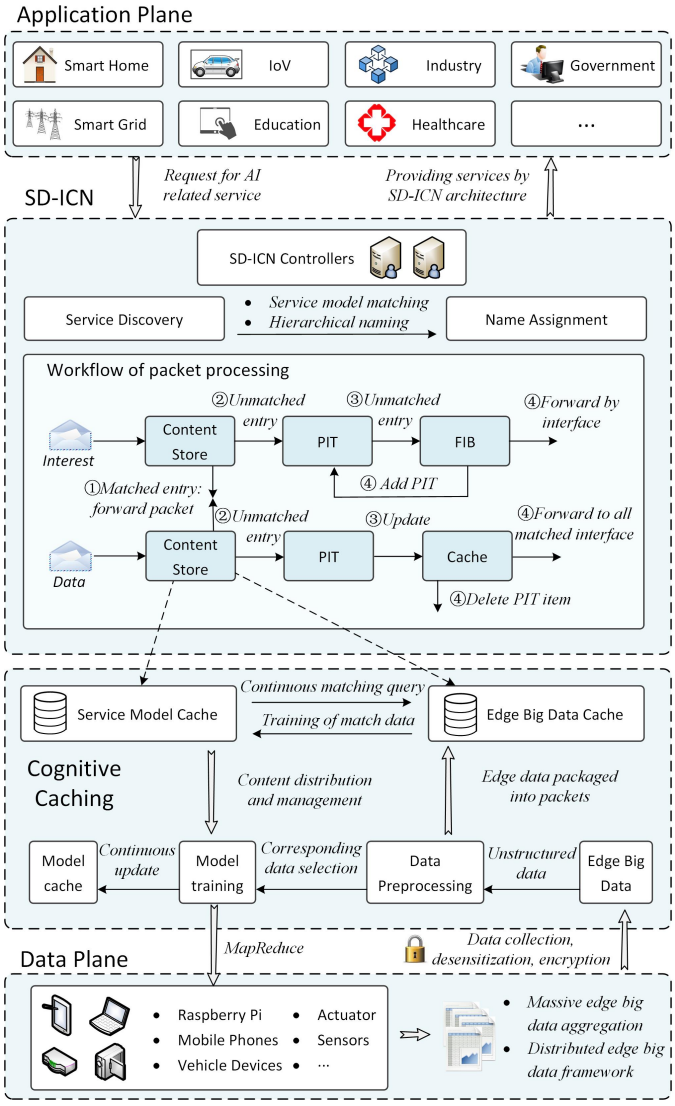


Fig. 2: Architecture of integrated decentralized big data and SD-ICN for AI service sharing.

the model is not available locally, the fog node broadcasts an interest packet to other nodes in the network, waiting for the returned information. If there is no ML model required by the service in the network, the local fog node determines whether to request the same type of data from other nodes according to the service precision requirement and the local data amount. If there is a large amount of relevant data locally, the fog node calculates the model locally or requests nodes with strong computing power nearby to complete the computing task. When there is not enough data locally to support the model training, the fog node broadcasts an interest packet and waiting for feedback from other nodes. The fog server therefore searches for workers as an edge master for AI model training in an Edge MapReduce (EMR) approach so as to make full use of computing resources at the edge of the network and achieve rapid model training. EMR implements reliability by distributing training of AI services to each available agent in Fog, and the agents periodically return the latest states of model training. With the advantage of SD-ICN, decentralized big data based artificial intelligence services can be efficiently and quickly distributed and cached at the edge of the network, alleviating the imbalance of computing resources and edge data. With the SD-ICN, the application of decentralized big data applications and AI service management is enhanced, providing a natural platform for edge intelligence.

### B. Basic Architecture of the SD-ICN for AI Sharing

Because of the advantages of ML in forecasting, decision making, perception and recognition, ML-based artificial intelligence services have penetrated into all areas of smart cities, including Industrial Internet of Things (IIoT), e-health, smart home, Intelligent Transportation Systems (ITS) and education. Due to the large scale of the aforementioned application scenarios, the same type of service is often requested by geographically distributed users. However, the existing traditional IP-centric networks are not efficient enough for high-speed content distribution [**?**]. When a certain user needs to continuously request a large amount of data, the traditional network is not sensitive to the specific content of the request. This has been significantly improved in the content-centric SD-ICN architecture. Frequent task requests and efficient content distribution need to be implemented by the caching and forwarding mechanism of SD-ICN.

In the proposed architecture shown in Fig.**??**, SD-ICN

directly interfaces with the needs of upper-level users. For the upper Service Layer, ML-based models and edge data are hierarchically named according to the service requests of users and applications. The name act as the vital part of interest packets and data packets in order to request relevant content from the SD-ICN. All nodes in the SD-ICN network follow the same set of naming and matching rules. The model that needs the data matches the corresponding packet name in the fog node, so the same type of data packet can be used for training of multiple models.

The SD-ICN Layer is the core for the implementation of AI sharing and distributing. When the fog node receives an interest packet of a certain kind of data or service, it first queries whether there is a corresponding content that can be matched in the local cache to be returned. If there is no matching local-cached content in the cache, the Pending Interest Table (PIT) is queried, and if it exists in the PIT, the PIT is updated and the packet is discarded. If the interest package PIT is not matched, continue to query the Forwarding Information Base (FIB). If the request can be matched in the FIB, the interest packet is forwarded to the destination node according to the defined forwarding rule and the interface, and the content of the PIT is updated. Otherwise, the interest packet is discarded. On the other hand, when the fog node receives the data packet, it also first queries the local cache. The packet is dropped if the packet already exists in the cache. Otherwise, continue to query the PIT. If the packet is matched in the PIT, the packet is cached and forwarded to all matching interfaces in the PIT, and the entry is subsequently removed from the PIT. If no match is found in the PIT, which means the data packet is irrelevant to the service or potential service of the node, and the data packet is discarded.

The Cognitive Caching ayer mainly implements model training and popularity-based cache. Massive edge data is aggregated in the fog node and preprocessed, structured, and secured. In addition to the current edge functions such as edge computing and device management, the fog nodes mainly act as the platform to manage the cached content and pre-processing of the data in the proposed architecture. The fog node can pre-process the data, perceive the local potential service, and name the data in the same hierarchical way as the service layer. For locally generated ML tasks, the fog node provides services to local users in the order of: 1) local model cache 2) broadcast model request 3) local data training 4) request data for training.

### C. Packet Definition in Proposed Architecture

The traditional ICN has only two kinds of packets, namely, interest packets and data packets. In order to adapt to the AI service sharing model proposed in this paper, this paper further subdivides the function of the data packet. Fig.**??** shows the four kinds of packets in the proposed architecture: Service Interest Packet (SIP), Data Interest Packet (DIP), Service Model Packet (SMP) and Data Packet (DP). We have defined the meaning of the fields in the packets as shown in Fig.**??**. Different from the traditional ICN, the cache space is divided into a data cache and a service model cache. They optimize

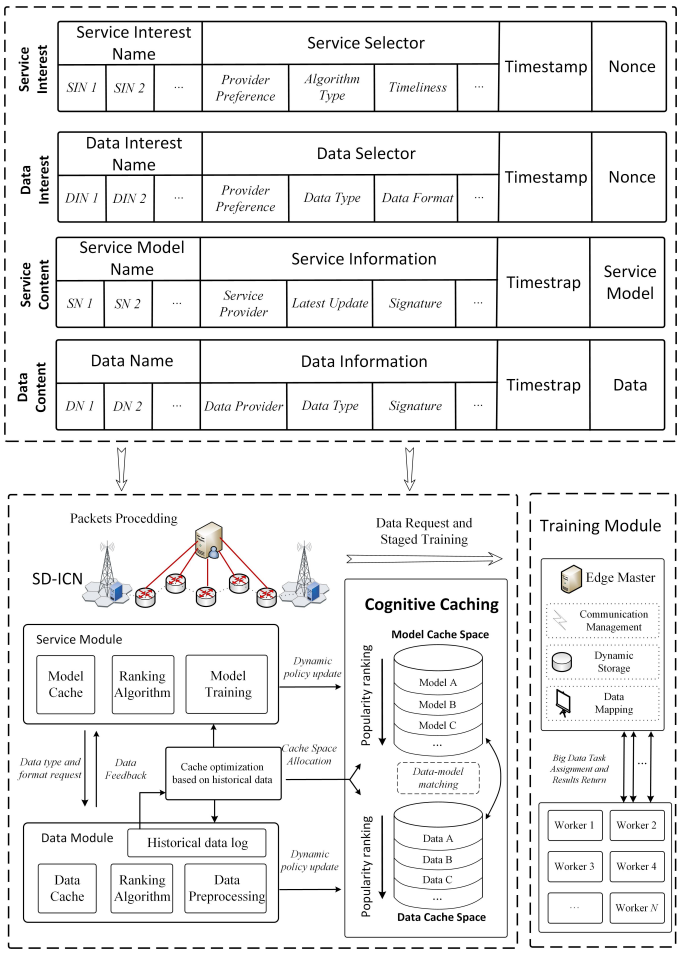


Fig. 3: Training, caching and distribution for AI service sharing by cognitive SD-ICN.

the cache based on the content popularity to ensure a higher cache hit rate. The fog node records the historical data of the request and response by the access log, and uses this as a basis to optimize the limited cache space. Each node optimizes its own cache based on its calculated popularity, to ensure that higher-popular content is more likely to be returned.

The specific functions of each type of data packet are as follows:

*1) Service Interest Packet (SIP)*: SIP is used to issue a request for a specific AI model. SIP includes requirements for service provider preferences, algorithm types, and the most recent update time of the model.

*2) Data Interest Packet (DIP)*: DIP is used to request a certain type of data for training the model. DIP mainly emphasizes various aspects of the requested data, including data type, format, provider, security, etc.

*3) Service Model Packet (SMP)*: The function of SMP is to return the pre-trained AI model. Different aspects of the information with the AI service are included in the header, including the source provider of the model, the updater, the update time, and the corresponding data type.

*4) Data Packet (DP)*: DP returns the specified type of data after preprocessing. The information in the DP header is mainly used to match the requirements of the DIP.