Data Mining and Analytics for Edge Caching

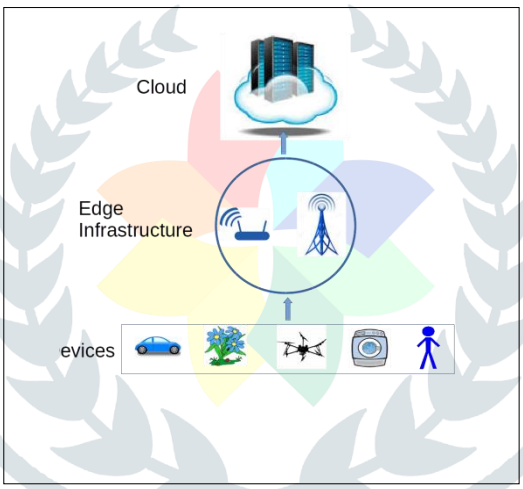
[1] Subhadra Kompella & [2] Shanti Chilukuri
[1,2]Department of Computer Science and Engineering, GITAM (deemed to be University), Visakhapatnam, India.

In view of the extensive usage of wireless networks for divergent applications in recent times, providing network solutions that result in good Quality of Experience (QoE) is a challenging task. Newer network proposals such as 5G networks have stringent network performance requirements even for dense networks with huge amount of traffic. Caching data on the network edge instead of the in the core network is a strategy that showed good promise towards meeting this go al. However, edge caching requires a good strategy that can suggest the location to cache and the content to be cached. Data mining and analytics techniques such as clustering and pattern recognition using data about the network such as the network usage patterns, resources and topology have been used successfully for effective edge caching. In this paper, we study the state-of-the-art that combines the two emerging and important techniques of edge caching and data mining and analysis for improved network performance, with the aim of identifying gaps in research in this area. We also put-forth the major challenges faced while using data mining and analysis techniques for edge caching.

Keywords: *wireless networks; edge caching; data mining; QoS.*

Introduction

The focus of the 5G specifications is to provide good Quality of Service (QoS) to challenging network environments such as Vehicular ad hoc networks (VANETs), Industrial Internet of Things (IIoT), Smart Cities etc. VANETs are increasingly popular due to interesting applications such as in-vehicle entertainment and information, obstacle detection, automatic cruise control etc. Smart Homes and Cities consist of applications like pollution monitoring, intruder detection, smart grids etc. IIoT includes fleet management, predictive maintenance, factory automation etc. Such applications typically gather data from or disseminate data in extreme network conditions such as poor links and high mobility rates that result in rapidly changing network topologies.



Several techniques have been suggested to improve the performance of 5G networks. Software Defined Networking (SDN) that separates the data and control planes of the network for better control and performance, edge computing, data analytics on the edge are among them. Traditionally, networks push the data to the cloud which resides several hops away from the producer/consumer of data. Pushing the data to the cloud where resources such as processing power, memory are abundant facilitates processing of data to gather meaningful information. However, the amount of data gathered in 5G networks is huge and the delay should be low (due to real time applications such as autonomous vehicles), in spite of the constantly changing network topology. Since the cloud is away from the producer/consumer, the traffic in the core network increases considerably due to processing in the cloud. Also, because the cloud is far away from the producer/consumer of data, there are considerable delays in pushing or pulling data to or from the cloud.

Edge computing stores and processes data on the network edge. This results in lesser traffic in the core network and also in reduced delay as the data is now closer to the producer/consumer. Because of this, edge computing has been proposed as a major technique to achieve the data rates and delays required by 5G network applications. Content Distribution Networks(CDNs), Mobile Edge Clouds and fog computing cache data on the edge. The most important decision parameters for edge caching are what to cache, where to cache and when to cache. Devices storing data on the edge are much more number but have limited resources compared to the cloud. Hence, while data processing in the cloud is mostly centralized with virtually unlimited resources, edge processing is distributed with limited resources. As such, the challenges faced by data processing on the edge are different from those for the cloud.

Data mining is a term that refers to the mechanisms for extracting meaningful information by analysis and interpretation of data. In the past one or two decades, data mining has been widely used to find patterns and analyze data for various applications such as financial markets, retail markets, network intrusion detection and management.

Data mining techniques have been used for edge data processing for content popularity prediction for caching, cleaning of data gathered before storing it at the edge/in the cloud networking etc. The confluence of data mining and

Big Data with the next generation networks (e.g., 5G networks) can be in two ways [1] –

Big Data driven networking in which the network is designed with flow and processing of Big Data in view and

Big Data assisted networking, where the network performance is improved based on data about the network resources, traffic etc.

The aim of this paper is to explore the state of the art in the second of the above two categories. Specifically, we focus on improvement of the network performance with edge caching, which in turn is facilitated by data mining techniques. We also identify some challenges of and open areas in data mining for effective edge caching.

Types of Edge Caching

Edge caching can be categorized in several ways. Two of the important classifications that we study are discussed below.

Based on infrastructure support necessary -

Edge caching may or may not basic network infrastructure. Accordingly, it can be classified into infrastructure-based and infrastructure-less edge caching.

- Infrastructure-based caching: The network has fixed base stations (BSs) cache data.
- Infrastructure-less caching: Caching is done at the device level. Based on

online or off line data analysis -

Edge caching benefits from data analysis. This data may be gathered from live data streams or from simulated networks.

- On-line data analysis: Analysis of the data may be online, where the data is cached on the edge depending on the data usage at that point of time.
- Off-line data analysis: The caching decision is made offline, based on data gathered previously.

Not all network scenarios may support a type of caching. For example, infrastructure-based caching is clearly not possible in ad hoc networks.

Related Work

[2] Studies the application of machine learning for edge caching. This work presents various techniques of Big Data analytics and machine learning for caching and processing data on the network edge. The authors identify data mining techniques such as pattern matching, text compression, clustering etc. for applications like intrusion and anomaly detection, self-organizing networks (SONs), automatic optimization of resources in the network, context-aware processing etc.

[3] Proposes data mining techniques for caching content on the edge. The authors present two caching strategies based on mining the interests of individual users or groups of users. They adapt a popular recommender technique called Matrix Factorization (MF) combined with the Least Recently Used (LRU) cache replacement strategy for edge caching. The resultant technique is called MFLRU, which is not very efficient if the interests of users or groups change slowly (in hours). For rapidly changing interest patterns, they propose a second technique based on collaborative filtering (K Nearest Neighbors or KNN) called KnnDyn. After calculating the group similarity at a central server that collects request information from edge servers, KnnDyn calculates the "*access score*" for each item in the cache. Replacement of cache items is done based on the access score. The authors evaluate the performance of the proposed strategies (in terms of the cache hit ratio) using a large, real-world request trace of a video-on-demand service.

[4] proposes a practical architecture that leverages on big data analytics for offloading data from the back haul links and achieving good Quality of Experience (QoE). The authors propose a distributed data storage platform such as Hadoop Distributed File System (HDFS) for collection of raw data at the base stations. The researchers collect users' mobile application data from a telecom operator from several base stations. After cleaning, parsing and formatting the data, both data and control packets are processed by applying data analytics techniques (Map-Reduce operations) on the header and payload information. In a practical case study, almost 80TB of network traffic is analyzed using Apache Hadoop. Their simulations show encouraging results depending on the storage size and availability of popularity data. The results presented by them are based on proactive caching of data, with nearly 98% of data offloaded from back haul links and almost 100% request satisfaction, depending on the available storage at the BSs. These results are with 13GB of data and 30% of content ratings and when data is cached at 16 BSs proactively. A similar study is done in [5] and upto 98% decrease in the backhaul traffic and 100% request satisfaction are reported for a different storage size (15.4GB), 10% content popularity rating availability and 16 BSs.

In [6], the authors explore the idea of live data analytics for collaborative edge and cloud rpocessing in wireless IoT. They present a framework for processing data in a coordinated manner by the edge and the cloud. IoT edge gateways store and process the raw data stream coming from different sensors and domains in semi-structured and unstructured forms. Since IoT devices are heterogeneous in nature with varying data properties with time and different IoT applications have different data dimensions, cloud computing is used to guide the working of the edge nodes.

[7] proposes Firework, a computing paradigm for Big data processing in collaborative edge environments (CEEs) for IoT. Data can be gathered from or sent to various types of stakeholders on heterogeneous platforms (Apache Spark, Apache Kafka, Raspberry Pi etc.). Firework creates a virtual view of the Distributed Shared Data (DSD) and creates abstractions for data sharing. The authors explore connected health and find-the-lost as two possible applications of

Firework. Privacy, programmability and extensibility of the paradigm are identified as challenges by the researchers.

In [8], the authors propose a Big Data deep reinforcement learning approach for Software-Defined Networks with mobile edge computing and caching. They use the learning mechanism for orchestrating networking, caching and computing resources dynamically in the context of Smart Cities. Wireless network visualization is done with a visualization hyper visor and an SDN controller is used to manage the caching and computing resources at network nodes. Resource allocation is modeled as an optimization problem that is solved using big data deep reinforcement learning. Implementation of the proposed method using TensorFlow [9] shows improved resource utilization with the proposed framework.

Caching on the edge is challenging in the context of a large number of mobile devices that use different apps. To facilitate faster data access in such scenarios, [10] gathers data from mobile devices to understand the temporal-spatial characteristics of app usage. They analyze this data and predict the top N app types under specific base stations (BSs). This is used for caching the data at the edge and the hit ratio is measured for a BS cluster. Upto 60% hit ratio is reported by the authors.

In [11], the authors transform the edge cache resource allocation problem into a multi-leader, multi- follower Stackelberg game, where the leaders are the users and the followers are the content providers (CPs). The combinations of the strategy combinations of the users and CPs and utility functions are analyzed. The authors prove that the Nash Equilibrium of the cache allocation among CPs exists and is unique for any strategy combination of the users. Simulation results show that their proposed algorithm results in better QoE with increased number of users, when compared with the minimum link cost algorithm.

A service delay minimization problem is formulated for small cell base stations in [12]. This optimization problem is divided into two sub-problems. Firstly, a clustering algorithm is proposed so that users with similar content popularity are associated with the same small cell base station where possible. Secondly, a learning algorithm is used to make the base stations learn the content popularity distribution for the groups of users serviced by them and cache data accordingly. Simulation results show lower average service delay and more offloading gain for varying cache sizes, when compared to random caching and learning-based caching without clustering.

In [13], the authors propose a deep reinforcement learning approach for integrated networking, caching, and computing for connected vehicles. They leverage on the principle of information centricity in Information Centric Networks (ICNs) and propose an integrated framework for the dynamic orchestration of network, cache and computing resources for different applications in vehicular networks.
This framework uses a Double-Dueling Deep-Q network algorithm to increase the utility of resources. A virtual BS is created at a BS or a Road Side Unit (RSU) whenever a vehicle requests for a video. The virtual BS acts as the site for edge caching. Simulation results show improved utility with varying parameters such as the content size, the price for MEC offloading, the price for accessing cache servers etc., when compared with similar schemes without MEC offloading and edge caching.

The authors of [14] propose to integrate a deep reinforcement learning method with federated learning to create an "In-Edge AI" framework. They use the deep reinforcement learning method to manage both the computing and communication resources. The deep reinforcement learning agents are trained using federated learning to reduce the data on the wireless up link, provide data privacy, adapt to heterogeneous user equipment and react dynamically to the ever-changing network environment. They capture Xender's trace of over 9000 users for a month and evaluate their framework. Results show improved cache hit-ratio, average utilities of user equipment and transmission cost.

In [15], a reinforcement learning method is used to provide secure mobile edge caching that is immune to jamming attacks. The authors identify specific challenges in providing security to mobile edge caching and propose reinforcement learning mechanisms for anti-jamming and light-weight authentication purposes. They also propose a secure collaborative caching mechanism using reinforcement learning.

There has been some work in the area of infrastructure-less edge caching. In [16], an energy efficient mechanism for content sharing in collaborative mobile clouds (CMCs) is proposed. Convex optimization is used for user scheduling and sub-channel assignment between base stations and mobile terminals to achieve energy efficient data delivery. Hypergraph coloring and multidimensional matching are used for efficient edge caching based on social ties and common interests in [17]. Evaluation of this scheme shows increased QoE using device to device (D2D) communication.

In [18], the authors propose a differentiated caching mechanism for vehicular networks. Assuming the information centric network model, they use a radial basis function for kernel ridge regression to dynamically allocate different parts of the available cache for satisfying the QoE of different vehicular network applications. Simulations results presented by them show that the proposed scheme results in decreased content retrieval time.

[19] proposes a Double Deep-Q network (Double DQN) learning framework for maximizing the cache hit ratio in hierarchical wireless networks. After a D2D sharing model is obtained, the authors formulate an NP-hard optimization problem. Then, they use the Markov Decision Process for cache replacement and also propose a caching strategy based on Double DQN. Parameters such as the hit ration, delay and cellular network traffic are evaluated based on data

captured from actual users. It is seen that the proposed caching algorithm results in better performance in terms of all these parameters.

The classification of the existing data analysis mechanisms for edge caching studied by us is given in Tables 1 and 2.

Infrastructure-based	Infrastructure-less
[3], [4], [5], [6], [7], [8], [10], [11], [12], [13], [14], [15]	[16],[17],[18],[19]

Table 1: Classification of existing work based on infrastructure support necessary

Online	Offline
[5], [13], [14], [16], [20], [17]	[1], [4], [11], [8], [10], [12], [18], [19], [21]

Table 2: Classification of existing work based on on-line/off-line data analysis Challenges

and Future Directions

The above discussion shows that while there are several data analysis techniques for edge caching in infrastructure-based networks, there are very few for infrastructure-less networks. In view of an increasing number of ad hoc network applications, better data analysis techniques for edge caching in such networks need to be explored. Also, it can be seen that there are more off-line techniques for data analysis. Since the network conditions and traffic patterns change rapidly in most network applications that benefit from edge caching (such as vehicular networks), it is imperative that more research is necessary for on-line data analysis.

In addition, some of the specific challenges that need focus are as below:

a) ***Data sparsity and density:*** In [2], the researchers mention that data sparsity and high density make it difficult to apply machine learning for challenging network scenarios. To overcome this, data should be processed before it is fed to the learning module. However, the limited resources combined with massive, high dimensional data at the edge make this challenging.

b) ***Privacy and security:*** Securing the network edge is much more than securing the cloud, because of the distributed nature of data. Providing security and privacy for large amounts of distributed data on the edge is a difficult task.

c) ***Time taken for analysis:*** While collaborative edge and cloud computing can benefit diverse IoT applications, IoT data is generally is bursty. In addition data gathered by sensors expires very quickly and has very small size. Hence, the raw data is processed at the edge and only meaningful information extracted from it is stored in the cloud. This requires efficient cooperative distributed processing by base stations and data compression. Processing of Big Data on the edge before the expiry of the data is achallenge in IoT networks, especially during the peak load time ([5]).

d) ***Energy efficiency:*** In the context of infrastructure-less networks with energy constraints like ad hoc sensor networks, data analysis that can be done with low energy consumption are necessary for on-line edge caching decisions.

Conclusions

In view of the increasing number of devices that vary in their capacities, edge caching has emerged as an interesting technique to provide the necessary user experience. However, there are several important factors that influence the performance of edge caching networks. Data mining and analysis methods such as clustering, machine learning and pattern recognition are being widely employed to make edge caching decisions that govern when, where and what to cache. In this paper, we study the existing techniques for data analysis for edge caching. We classify the existing solutions into infrastructure-based or infrastructure-less solutions and on-line or off-line solutions with a view to identify the gaps in this area. We further point to some challenges in this area that can be directions for future research.

References

[1] N. Zhang, P. Yang, J. Ren, D. Chen, L. Yu and X. Shen, "Synergy of Big Data and 5G Wireless Networks: Opportunities, Approaches, and Challenges," in IEEE Wireless Communications, vol. 25, no. 1, pp. 12-18, February 2018.

[2] Z. Chang, L. Lei, Z. Zhou, S. Mao and T. Ristaniemi, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," in IEEE Wireless Communications, vol. 25, no. 3, pp. 28-35, JUNE 2018.

[3] Guangyu Li, Qiang Shen, Yong Liu, Houwei Cao, Zifa Han, Feng Li, and Jin Li. "Data-driven Approaches to Edge Caching", in Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies (NEAT '18). ACM, New York, NY, USA, pp. 8-14, 2018.

[4] E. Zeydan et al., "Big data caching for networking: moving from cloud to edge," in IEEE Communications Magazine, vol. 54, no. 9, pp. 36-42, September 2016.

[5] E. Baştuğ et al., "Big data meets telcos: A proactive caching perspective," in Journal of Communications and Networks, vol. 17, no. 6, pp. 549-557, Dec. 2015.

[6] S. K. Sharma and X. Wang, "Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks," in IEEE Access, vol. 5, pp. 4621-4635, 2017.

[7] Q. Zhang, X. Zhang, Q. Zhang, W. Shi and H. Zhong, "Firework: Big Data Sharing and Processing in Collaborative Edge Environment," 2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, 2016, pp. 20-25.

[8] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung and H. Yin, "Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," in IEEE Communications Magazine, vol. 55, no. 12, pp. 31-37, Dec. 2017.

[9] Abadi, Martin & Agarwal, Ashish & Barham, Paul & Brevdo, Eugene & Chen, Zhifeng & Citro, Craig & Corrado, G.s & Davis, Andy & Dean, Jeffrey & Devin, Matthieu & Ghemawat, Sanjay & Goodfellow, Ian & Harp, Andrew & Irving, Geoffrey & Isard, Michael & Jia, Yangqing & Kaiser, Lukasz & Kudlur, Manjunath & Levenberg, Josh & Zheng, Xiaoqiang. . "TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems." Technical Report, 2015.

[10] M. Zeng et al., "Temporal-Spatial Mobile Application Usage Understanding and Popularity Prediction for Edge Caching," in IEEE Wireless Communications, vol. 25, no. 3, pp. 36-42, JUNE 2018.

[11] Z. Zheng, L. Song, Z. Han, G. Y. Li and H. V. Poor, "A Stackelberg Game Approach to Proactive Caching in Large-Scale Mobile Edge Networks," in IEEE Transactions on Wireless Communications, vol. 17, no. 8, pp. 5198-5211, Aug. 2018.

[12]M. S. ElBamby, M. Bennis, W. Saad and M. Latva-aho, "Content-aware user clustering and caching in wireless small cell networks," 2014 11th International Symposium on Wireless Communications Systems (ISWCS), Barcelona, 2014, pp. 945-949.

[13] Y. He, N. Zhao and H. Yin, "Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 67, no. 1, pp. 44-55, Jan. 2018.

[14] Wang X, Han Y, Wang C, Zhao Q, Chen X, Chen M. In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. arXiv preprint arXiv:1809.07857. Sep 2018.

[15] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen and M. Guizani, "Security in Mobile Edge Caching with Reinforcement Learning," in IEEE Wireless Communications, vol. 25, no. 3, pp. 116-122, JUNE 2018.

[16]Z. Chang, J. Gong, T. Ristaniemi and Z. Niu, "Energy-Efficient Resource Allocation and User Scheduling for Collaborative Mobile Clouds With Hybrid Receivers," in IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9834-9846, Dec. 2016.

[17]B. Bai, L. Wang, Z. Han, W. Chen and T. Svensson, "Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework," in IEEE Wireless Communications, vol. 23, no. 4, pp. 74-81, August 2016.

[18] V. S. Varanasi and S. Chilukuri, "Adaptive Differentiated Edge Caching with Machine Learning for V2X Communication," 2019 11th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 2019, pp. 481-484.

[19] Wenkai Li, Chenyang Wang, Ding Li, Bin Hu, Xiaofei Wang, and Jianji Ren, "Edge Caching for D2D Enabled Hierarchical Wireless Networks with Deep Reinforcement Learning," Wireless Communications and Mobile Computing, vol. 2019, Article ID 2561069, 12 pages, 2019.