Distributed Data Analytic Models for IoT Edge Computing Network

Tony Jan, Savitri Bevinakoppa
School of Information Technology and Engineering
Melbourne Institute of Technology
288 La Trobe St, Melbourne, VICTORIA 3001
AUSTRALIA
tjan@mit.edu.au, sbevinakoppa@mit.edu.au http://www.mit.edu.au

Abstract: - This paper introduces a distributed machine learning model for response control at the IoT edge computing network. The resource-limited IoT edge devices are pooled together for a distributed processing in void of remote cloud servers. The edge computing offers prompt and comprehensive response in mission critical applications. This paper addresses the current and future challenges and opportunities in real-time IoT services and applications. Further, this paper also gives number of platforms and user requirements in selecting platforms.

Key-Words: - IoT, Data analytic, distributes network, edge computing

1 Introduction

The massively interconnected cyber-physical devices with limited computing capacity are aggregated to form the Internet of Things (IoT) and they are expected to provide smart applications in various industries such as hospital care and public infrastructure protection [1].

The IoT devices, despite their initial purpose of smart home automation with no control of Quality of Service (QoS), they are now deployed in mission critical applications requiring real-time QoS requirements.

This new expectation is creating a dilemma in the current IoT architecture in which Cyber-Physical Devices (CPD) heavily rely on cloud-based Decision Support Systems (DSS) for data storage and analytic processing. The heterogeneous, dynamic, and massive IoT network is generating enormous amount of data for analytic processing; increasing its reliance on the remote cloud server even more.

The CPDs and DSS are usually not co-located, and they can be, at times, located in different continents. The DSS can send back the response messages with high priority, but the response delay is a major hindrance to the deployment of IoT services in the mission-critical real-time applications [2].

For proactive and faster responses to the critical incidents in hospital care, nursing-home care, outback bush fire, and policing acts, we need to be able to embed data analytic and decision support capacity in the CPDs themselves so that they can bypass the use of remote DSS; allowing faster edge computations for real time processing and response as shown in Figure 1.

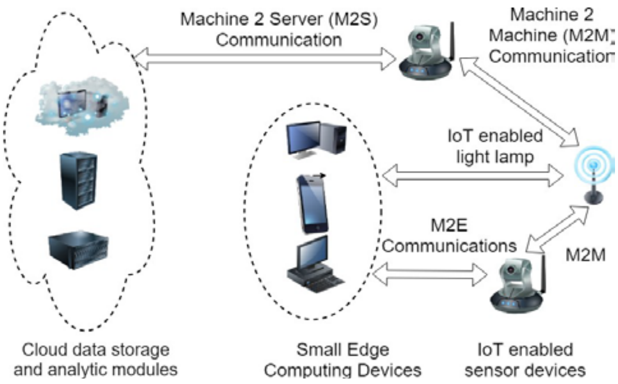

Figure 1: IoT Architecture

The challenge remains on how to embed data analytic capacity in small CPDs where the computing resource is severely limited. The expectation is the clustered small IoT-CPDs are interconnected in a way to accommodate the data analytic processing in a distributed mode [3].

In this paper, we review the current practices of distributed machine learning approach for faster real-time response. In particular, this paper proposes an innovative data analytic model that is adjusted for distributed processing over the IoT edge networks. This paper compares the distributed data analytic models under the stringent requirements of IoT edge networks. The paper further reviews appropriate technologies for real-time IoT network applications.

2 Literature Review

This section explains literature review of IoT and processing power.

2.1 Review of IoT Edge Computing

There has been some significant research work on embedding data analytic modeling in the edge IoT devices for proactive critical responses [4]. The edge computing is described in Figure 2.

The IoT edge computing is mainly used for computational offloading of data storage and analytic processing. The offloading must take into consideration of dynamic nature of network access requirements, number of edge devices, and computational resources of the edge devices. We must take into consideration of the granularity and hierarchy of edge network topology and how to dynamically partition the application for offloading [5].

MAUI [6] offers code offloading for adaptive decision making on network availability but it requires manual annotation of the offloading parts. In dynamic IoT edge network, manual annotation may be a limiting factor.

COMET [7] offers virtual machine synchronization and shared memory to adapt the edge devices to be part of the computational network.

Think Air [8] offers parallel virtualization over the small edge computing devices to achieve sufficient computing power for intelligent data analytic processing.

Kang et al [9] offers random allocation of resources followed by iterative redistribution of resources according to the dynamic network and user behaviour. The analysis of network and user behaviour in IoT edge network may be computationally intensive.

Fan et al [10] offers an inter data centre based load redistribution for efficient energy usage and reduced latency. Although their idea is meant for multi data centres, some of their systematic approach can be useful for edge computer sharing.

Nishio et al [11] introduces mobile cloud in which small portions of processing tasks are distributed amongst small mobile devices under the direction of supervisory control module in awareness of latency and resource optimization.

The underlying idea is to divide the processing task into small modules for distributed processing 'in awareness of latency and resource optimization' amongst the myriad of small mobile devices as shown in Figure 2.

The challenge remains on how to segment the data analytic processing tasks into the small modules for distributed processing.

In the next section, we review the appropriate data analytic models for distributed computing. We also propose an innovative ensemble of linear regression models as part of the solution to the distributed IoT edge computing. The proposed model is compared to the other state-of-the-art models in a simple experiment of location-aware IoT service application.

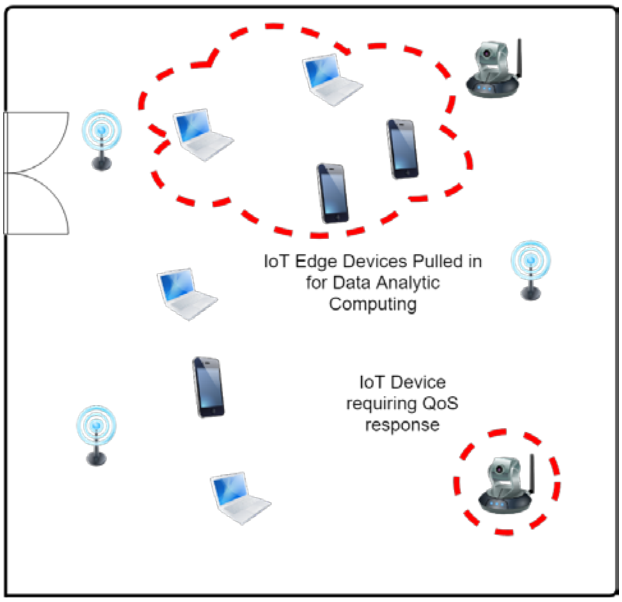

Figure 2: IoT Edge Computing

2.2 Review of Distributed Machine Learning Models

Data analytic computing is inherently intensive, and it may require distributed processing [12].

A well-known heuristic model such as Multi-Layer Perceptron (MLP) is known for deep learning through multiple layers of interconnected intricate memory modules. MLP is not suitable for distributed processing as the hierarchical segmentation of the network itself is a challenge. The reassembly of the outputs from the segmented components may not reflect the true learning of the global MLP [13].

Another popular model such as Self Organizing Map (SOM) is useful in uncovering data patterns. An advanced variant of SOM has been applied to IoT resource scheduling and sharing with stringent restriction on energy consumption. This work is appropriate for cloud-server based processing but not suitable for the edge-based processing because the sample data points increase exponentially in predictive modeling of IoT network usage requirements, capacities, and availabilities [14].

The other popular models such as Support Vector Machine (SVM) and its advanced variant such as knn-SVM-PSO [15] are the exciting advancement; however, again, they are developed for remote cloud server processing, not for the resource limited IoT edge devices [16].

Given the challenge of the resource-limited IoT edge devices, an intuitive solution is to use an ensemble of simple (data analytic) models. In such case, each data analytic component is already well segmented from the inception; and there are more options to intelligently combine the outcomes from the disjoint learning modules for optimal global optimization. In particular, we take a great interest in Adaboost ensemble of linear regression models for IoT edge computing. This model is shown to outperform other complex nonlinear regression models in dynamic modeling applications [17]. In the following section, we examine the aforementioned model and adjust its architecture to make it suitable for IoT edge computing. The adjusted model is discussed in detail and compared against the other related state-of-the-art models in a simple experiment.

3 Proposed Model

The proposed model is based an Adaboosted Linear Regression (ALR) which can achieve a good trade-off between bias and variance given limited computational resources [18]. The ALR is well-known in machine learning community but has not been exclusively applied in the IoT edge distributed computing. The learning model architecture is described in Figure 3.

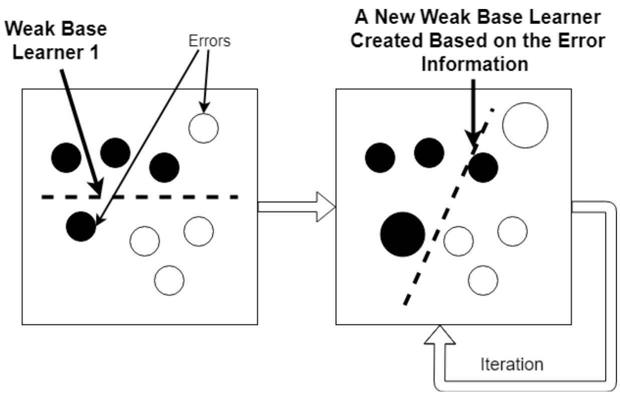

Figure 3: Adaboot Learning

The first weak base learner is designed based on training sample. The data points in the error region(s) are given more weight in the next iteration to create a new weak base learner. The final model is an intelligent weighted combination of weak linear regression models as shown in Figure 4.

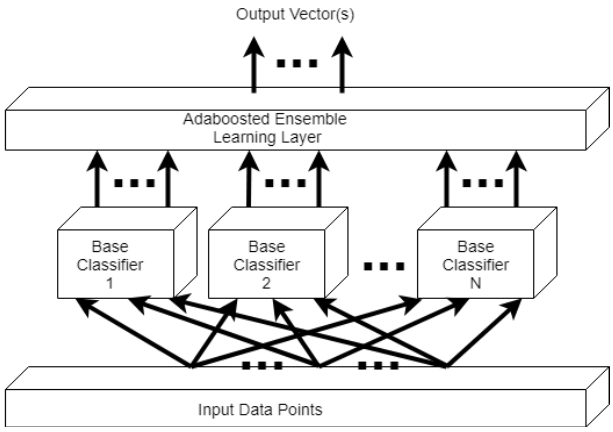

Figure 4: Ensemble Model

Each base (linear regression) model can be processed on a simple IoT edge device and its outputs can be readily assembled back by a simple weight function. In our proposed model, the simple weight function is adjusted to include kernel based smoothing function to cater for global optimization. The simple change is that the outputs from the base models are better combined with easier control with a single smoothing parameter.

The final model is an adjusted Adaboost ensemble of linear regression models which can mimic far more complex nonlinear regression models. Such a model is suitable for distributed computing as shown in the next section.

4 Experimental Analysis

IoT latency requirement becomes more obvious in the real-time location-based applications. For this aim, we use the bench mark data from the location sensing application with signals received from the array of iBeacons [19].

The dataset was created using the RSSI readings of an array of 13 iBeacons in the first floor of Waldo Library, Western Michigan University. Data was collected using iPhone 6S. The instances) and an unlabeled dataset (5191 instances).

The experiment includes a mobile phone user roaming through the floor of library and the data analytic models must be able to determine the location of the user in real time. The computational resources are restricted to the available devices in the proximity pooled in for edge computing.

Given such a restricted resource condition, several data analytic models are deployed to detect the location of the mobile user. The performance of each analytic model (under the restricted computing environment) is recorded and compared. The Preliminary experimental outcomes show the proposed linear model achieves improved data analytic performance given much reduced computational resources, as expected. When computational resources increase, the nonlinear regression models (such as neural network or SVM) would outperform our linear model; however, given the limited computational resources at the IoT edge devices, the proposed model shows better modeling performance than the other data analytic models.

Give the limitations of IoT edge network devices, the proposed model can be useful to provide real-time data analytic processing for real-time responses and QoS controls.

5 Platforms to optimize edge-based data analytics

This section provides platforms to optimize edge-based data analytics and users' requirements of IoT edge platforms [20, 21, 22].

5.1 Platforms to optimize edge-based data analytics

Five important platforms that can be used to optimize edge-based data analytics [20] shown in Table 1.

Table 1: Platforms to optimize edge-based data analytics

Platform	Features
Azure stream	- Microsoft IoT platform - Real-time analytical solution - Can handle complex event processing - Real-time dashboard display
IBM Watson	- Built on Apache Edgent - IBM provides processing, analytical power and machine learning at the edge of the network
Cisco connected streaming analytics	- Supports high velocity data streaming from multiple sources - Suited to continuous monitoring of live data streams
Oracle edge analytics	- Processes data on embedded devices - Support downstream applications, providing real-time analytics - High processing speed and real time data capture capabilities
Intel analytics toolkit	- Collects data from smart sensors - Requires little storage and processing capacity - Real-time, cost-effective and powerful way

5.2 User requirements of IoT edge platforms

Features and capabilities that most of the users require to use IoT edge platforms based on users' experiences [21] are

1. Extensive protocol support for various data input
2. Robust capabilities even for offline functionality
3. Cloud based support
4. Support of scalable architecture
5. Comprehensive analytics and visualization tool

6 Conclusion

The paper reviews the state-of-the-art distributed computing architecture for data analytic processing given very limited pool of computational resources. The results show that Adaboot ensemble of linear regression model with adjustable smoothing parameter has shown comparable modeling accuracy under limited computational complexity. The experimental outcome is also positive requiring extensive testing on a large-scale dataset. This paper gives overview of edge-based IoT platforms and user requirements. Future work will be on comparative study of platforms with real world case studies.

References:

[1] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, pp. 431–440, 2015.

[2] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in Proceedings of the 2015 workshop on mobile big data, 2015, pp. 37–42.

[3] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1185–1192, 2017.

[4] N. I. M. Enzai and M. Tang, "A taxonomy of computation offloading in mobile cloud computing," in Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014

2nd IEEE International Conference on, 2014, pp. 19–28.
[5] E. D. Carreno and P. O. Navaux, "IoT Workload Distribution Impact Between Edge and Cloud Computing in a Smart Grid Application," in High Performance Computing: 4th Latin American Conference, CARLA 2017, Buenos Aires, Argentina, and Colonia del Sacramento, Uruguay, September 20-22, 2017, Revised Selected Papers, vol. 796, 2017, p. 203.
[6] E. Cuervo, A. Balasubramanian, D. ki Cho, A. Wolman, S. Saroiu,
R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in Proceedings of the 8th international conference on Mobile systems, applications, and services, 2010, pp. 49–62.
[7] M. S. Gordon, D. A. Jamshidi, S. A. Mahlke, Z. M. Mao, and X. Chen, "COMET: Code Offload by Migrating Execution Transparently." in OSDI, vol. 12, 2012, pp. 93–106.
[8] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in Infocom, 2012 Proceedings IEEE, 2012, pp. 945–953.
[9] Y. Kang, Z. Zheng, and M. R. Lyu, "A latency-aware co-deployment mechanism for cloud-based services," in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, 2012, pp. 630–637.
[10] Y. Fan, J. Chen, L. Wang, and Z. Cao, "Energy-Efficient and Latency Aware Data Placement for Geo-Distributed Cloud Data Centers," in International Conference on Communications and Networking in China, 2016, pp. 465–474.
[11] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in Proceedings of the first international workshop on Mobile cloud computing & networking, 2013, pp. 19–26.
[12] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," Neuro computing, vol. 237, pp. 350–361, 2017.
[13] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase,
[14] C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in Networks, Computers and Communications (ISNCC), 2016 International Symposium on, 2016, pp. 1–6.
[15] N. Abuzainab, W. Saad, C.-S. Hong, and H. V. Poor, "Cognitive hierarchy theory for distributed resource allocation in the internet of things," arXiv preprint arXiv:1703.07418, 2017.
[16] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," Applied Soft Computing, vol. 38, pp. 360–372, 2016.
[17] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," Expert Systems with Applications, vol. 67, pp. 296–303, 2017.
[18] M. Yousefi, M. Yousefi, R. P. M. Ferreira, J. H. Kim, and F. S. Fogliatto, "Chaotic genetic algorithm and Adaboost ensemble metamodeling approach for optimum resource planning in emergency departments," Artificial intelligence in medicine, vol. 84, pp. 23–33, 2018.
[19] S. O. M. Kamel, N. H. Hegazi, H. M. Harb, A. S. T. E. Dein, and H. H.
[20] Accessed on 26/8/18 https://www.networkworld.com/article/3206727/internet-of-things/5-of-the-best-data-analytics-platforms-that-use-the-edge.html
[21] Accessed on 26/8/18 https://www.networkworld.com/article/3247801/internet-of-things/the-top-5-user-requirements-of-iot-edge-platforms.html
[22] E. Siow, T. Tiropanis, W. Hall, Analytics for the Internet of Things: A Survey, ACM Computing Surveys, Volume 51 Issue 4, August 2018