# A Performance Evaluation of General Traffic Systems by Machine Learning

Suguru Nii and Takashi Okuda
Department of Information Science and Technology
Aichi Prefectural University
Aichi Japan
Email: {is141057@cis, okuda@ist}.aichi-pu.ac.jp

***Abstract*—When we design the Internet of Things (IoT) systems, it is important to evaluate general traffic systems. However, in general traffic systems, the exact solution is not available. Alternatively, we can evaluate it with simulation. However, simulation spends much time. In this paper, we propose evaluate general traffic systems by machine learning.**

## I. Introduction

As we are moving towards the Internet of Things (IoT), many "things" are connected to the Internet [1]. Cisco Systems, Inc. estimates 5 billion things will be connected to the Internet by 2020 [2].

IoT data that come from "things" are huge and many kinds. Therefore, IoT data arrive bursty at IoT Data Processing Systems (IDPS) that process IoT data. In addition, IoT data size is many kinds. Therefore, it is important to evaluate general traffic systems, which distribution of arrival and service are general (GI and G) and the number of servers is s (GI/G/s systems).

However, the exact solution of GI/G/s systems is not available [3]. Alternatively, we can use simulation or Queueing Tables [4] (tabulate from simulation results). However, not all of GI/G/s systems can be evaluated due to limitations of the data of Queueing Tables. In addition, simulation spends much time.

In this paper, we propose evaluate GI/G/s systems by machine learning. The goal of us is to evaluate any GI/G/s systems quickly and precisely. In addition, revealing what kind of teacher data we should use.

The organization of this paper is as follows: Section II states assuming IDPS. Section III states a performance evaluation of GI/G/s systems by machine learning. Section IV presents and discuss numerical results. Finally, section V summarizes the conclusion of this paper.

## II. Assuming IoT Data Processing Systems

Fig. 1 shows assuming IDPS. We model IDPS as G/G/s systems that process data from IoT device. IoT data arrive at IDPS with rate $\lambda$ which distribution is determined by squared coefficient of variation (SCV) $Ca^2$. The data is processed with rate $\mu$ which distribution is determined SCV $Cs^2$. The order of processing is First-In First-Out (FIFO). The number of servers is $s$. All of Servers performance are uniform. Buffer size is finite or infinite. $Ca^2$, $Cs^2(C^2)$ = 0 implies a deterministic distribution, 0 < $C^2$ < 1 implies a Erlang-$k$ distribution, $C^2$ = 1 implies an exponential distribution, $C^2$ > 1 implies a hyper-exponential distribution [5].

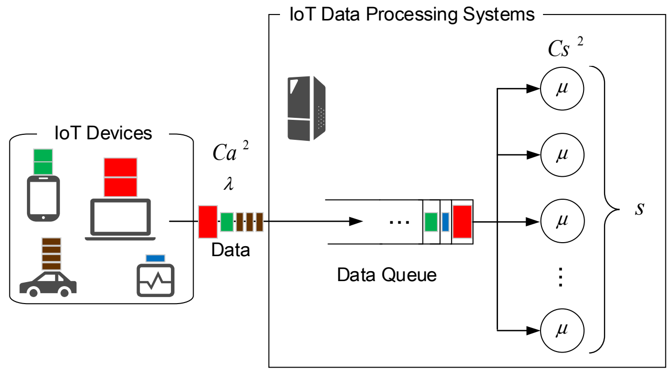


Fig. 1. IoT Data Processing Systems

## III. A Performance Evaluation of GI/G/s Systems by Machine Learning

In this section, we explain about a performance evaluation of GI/G/s systems by machine learning. We use Neural Network (NN) as machine learning. The reason why we use NN is development environment is fully regulated and processing is speedy [6].

The procedure of a performance evaluation of GI/G/s systems is as follows. Fig. 2 shows the flow of performance evaluation.

Step1: Make teacher data.
Step2: Construct a NN.
Step3: Make NN learn by using teacher data.
Step4: Input parameters, which define GI/G/s systems, to NN and evaluate the performance.

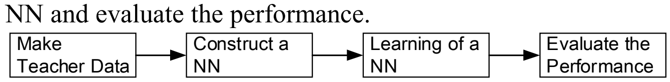


Fig. 2. The flow of performance evaluation

Here teacher data have component (1) and component (2). Component (1) consists of parameters and evaluation value of particular GI/G/s systems that have exact solution [7]. Component (2) consists of parameters and evaluation value of GI/G/s systems that get from simulation. We do not use component (1) and component (2) simultaneously.

## IV. Numerical Example

We evaluate *expected time in the systems* $W$ of GI/G/s

TABLE II
VALUES THAT USE WHEN WE MAKE TEACHER DATA

| Teacher Data | Queueing Systems | $Ca^2$ | $Cs^2$ | $\lambda$ | $\mu$ | $s$ | $\rho$ |
|---|---|---|---|---|---|---|---|
| Component (1) | M/G/1 | 1.0 | 0.1, 0.5 | 0.1, 0.3, 0.5 | 1.0 | 1 | 0.1, 0.3, 0.5 |
| | GI/M/s | 0.1, 0.5 | 1.0 | 0.1, 0.2, 0.3, 0.5, 0.6, 1.0, 3.0, 5.0 | | 1,2,10 | |
| Component (2) | GI/G/s | 0.1, 0.5, 1.0 | | | | | |

TABLE I
PARAMETERS OF GI/G/S SYSTEMS THAT WE WILL EVALUATE

| Items | Symbols | Numbers |
|---|---|---|
| SCV | $Ca^2, Cs^2$ | 0.1, 0.25, … 1.0, 2.5, … , 10.0 |
| Arrival rate | $\lambda$ | 8.0 |
| Service rate | $\mu$ | 1.0 |
| The number of clouds | $s$ | 10 |
| Utilization rate | $\rho$ | 0.8 |

systems that have parameters we show in Table I. Now $\rho(=\lambda/s\mu)$ means the utilization rate. In addition, buffer size is infinite.

NN is consists of 3 layers: input, hidden, output. The number of neurons of each layer is 4, 5, 1. 4 neurons of input layer correspond with $Ca^2$, $Cs^2$, $s$, $\rho$, 1 neuron of output layer corresponds with $W$. We use chainer [8] of libraries on an implementation of NN.

Component (1) uses the value of Table II and exact solution $W$ by using it. Component (2) also uses the value of Table II and evaluation value $W$ of simulation by using it. Simulation is executed 1 million times for getting one of the $W$. We use discrete-event simulation package CSIM20 [9] when we make component (2) by simulation.

The number of learning of NN is 6000 times enough to reduce losses.

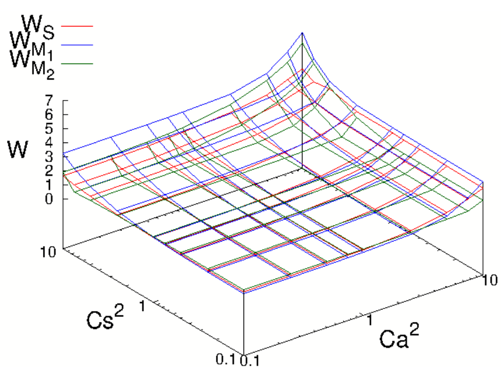


Fig. 3. Expected time in the systems

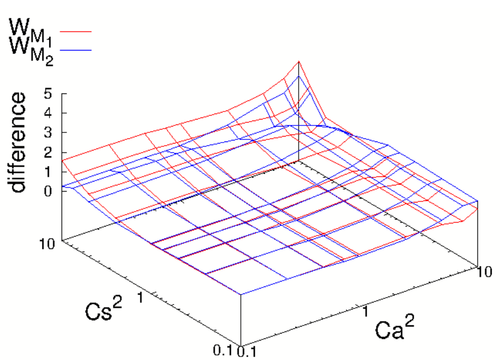


Fig. 4. The difference between $W_{M_n}$ and $W_S$

Fig. 3 shows the $W$ of GI/G/s systems. In the Fig. 3, estimation values of $W$ that we got from learned NN by using component (1) and component (2) are represented as $W_{M_1}$ and $W_{M_2}$. To compare the $W_{M_n}$ ($n$ = 1, 2), we show true values $W$ ($W_S$) that got from only simulation (not using machine learning). The $x$, $y$, $z$ axis show SCV $Ca^2$, $Cs^2$ and $W$.

First, we consider the relation of $W_{M_n}$ and $W_S$. $W_{M_n}$ is almost same value as $W_S$.

Next, we show the difference between $W_{M_n}$ and $W_S$ in the Fig. 4. The $x$, $y$ axis are same as Fig. 3. $z$ axis show the difference between $W_{M_n}$ and $W_S$. The difference of $W_{M_2}$ is smaller than that of $W_{M_1}$. Parameters of component (2) can be set more flexible than component (1). Probably, this is the reason why we could get this result.

Finally, we measure the calculation time that we need to complete a performance evaluation of GI/G/s systems. The PC spec we used is OS:Linux, CPU:Intel Xeon 2.7GHz, Memory:16GB. From Table III, the calculation time of $W_{M_1}$ is less than 20 seconds. In addition, the calculation time of $W_{M_2}$ is less than half of that of $W_S$. From the above, if Component (2) is used as teacher data, we can evaluate a performance of GI/G/s systems quickly, and precisely.

TABLE III
CALCULATION TIME

| | $W_S$ | $W_{M_1}$ | $W_{M_2}$ |
|---|---|---|---|
| Making teacher data | - | 0s 31 | 36s 71 |
| Learning | - | 16s 96 | 24s 20 |
| Sum | 153s 75 | 17s 27 | 60s 91 |

s = second

## V. CONCLUSION

In this paper, we evaluate a performance of GI/G/s systems by machine learning. As a result, we can evaluate them quickly and precisely. We will evaluate a performance of Fog Computing systems that is paid attention recently.

## REFERENCE


[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", IEEE Communication Surveys & Tutorials, vol. 17, no. 4, 2015.

[2] Cisco Systems, "Fog Computing and the Internet of Things:Extend the Cloud to Where the Things Are". [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. [Accessed: 2018].

[3] M. Miyazawa, *Mathematical Modeling for Queues and its Applications*, Makino Bookstore, 2013.

[4] L.P. Seelen, H.C. Tijms, M.H. Van Hoorn, *Tables for Multi-Server Queues*, North-Holland, 1985.

[5] H. Akimaru, K. Kawashima, *Teletraffic Theory and Applications*, OHM Co,. Ltd, 1993.

[6] H. Ichikawa, *Hierarchical Neural Network: Application to Non-Linear Problem*, Kyoritsu Publication, 1993.

[7] T. Okuda, M. Anthony, Y. Tadokoro, "A Performance Evaluation for General Traffic Model GI/G/s by neural network", IEICE Trans. B-I, vol. 76-B1, no. 10, pp. 730-733, 1993.

[8] Preferred Networks, https://chainer.org/, [Accessed: 2018]

[9] Mesquite Software, http://www.mesquite.com/. [Accessed: 2018]