

A Study of Throughput Prediction using Convolutional Neural Network over Factory Environment

Yafei Hou^{*†}, Kazuto Yano[†], Norisato Suga^{‡†}, Julian Webber^{§†}, Eiji Nii[†], Toshihide Higashimori[†],
Satoshi Denno^{*†} and Yoshinori Suzuki[†]

^{*}Graduate School of Natural Science and Technology, Okayama University, Okayama City, Japan

[†] Wave Engineering Laboratory, ATR International, Seika-cho, Soraku-gun, Kyoto, Japan

[‡]Faculty of Engineering, Tokyo University of Science, Tokyo, Japan

[§]Graduate School of Engineering Science, Osaka University, Toyonaka City, Japan

Contact Author: yfhou@okayama-u.ac.jp

Abstract—In this paper, using the time-series throughput data generated from a simulated factory scenario, we study throughput prediction using convolutional neural network (CNN). Different with image or numerical recognition using CNN, in which the distribution of the prediction target during training stage usually has the similar level, the distribution of the throughput data concentrates only on several values. This centralized distribution may degrade the prediction accuracy. Therefore, we will propose a new CNN prediction method employing target vectorization which can mitigate the centralization of distribution. This method makes training process of CNN hold more possibility and improves the prediction accuracy of the throughput.

I. INTRODUCTION

The era of Internet of things (IoT) and machine learning is coming which will be a boom for increasing smart usage in manufacturing sites such as factories and warehouses [1]. However, due to the special environment such as existence of many metal machines or mobile robots, the quality of a wireless communication link dynamically changes and some unexpected troubles on wireless communication occur which severely degrade the productivity of a factory. To maintain the reliability of wireless communication and avoid the unexpected loss from a bad wireless link, it is important and beneficial to predict future quality of service (QoS) of a wireless link such as throughput and latency to avoid serious damage to the productivity.

In addition, the correct prediction of wireless communication quality can also benefit the scheduling of MAC-layer and power control in the physical layer. For some scenarios such as in factories, many robots such as automated guided vehicle (AGV) and stacker crane move along the predetermined route [2]. These moving robots are usually connected and controlled via wireless communication because it is impractical to use a wired link. Nowadays, these robots are becoming prevalent in the factory environment, and thus the prediction of wireless communication quality such as throughput of these robots is highly important.

For QoS prediction, some tutorial papers [3, 4] have summarized most of the existing methods for the optimization

of wireless resource allocation. In addition, during the past decade, many researches try to use machine learning (ML) for the QoS prediction and many ML-based algorithms have been proposed and compared [5], such as using support vector regression for the prediction of TCP throughput [6] and using (F)ARIMA model for the prediction of HTTP traffic in [7]. We recently predicted the future received signal strength (RSS) variation using a probabilistic neural network (PNN) in [8]. Generally, ML-based methods can predict future events accurately when the input pattern is similar to the pattern that has been already observed in the past. However, the prediction accuracy of the ML-based methods usually degrades for some rare/untrained inputs.

In factories and warehouses, many robots move along the predetermined route. Therefore, their RSS data usually appears some periodic patterns for learning and prediction. Compared with RSS prediction, throughput prediction is more important and changeable for real-time application processed at IoT devices or robots such as monitoring video camera, audio or file upload service. In factories and warehouses, although there is a demand for wireless connection between the IoT devices or robots to networks, radio interference and RSS variation due to movement of robots and/or obstacles in a multi-path environment make a wireless communication link unstable. In such case, the correct throughput prediction will benefit to secure the important wireless application from serious damages to the productivity.

The time-series data prediction can be operated using different ML-based methods. Convolutional neural network (CNN) is largely used for the time-series data prediction which can process data before training to find major patterns of time-series data. CNN has become favorable for extracting features from time series signals, and therefore for detection, prediction, and classification. The essential part of this approach is that data cleaning is not required, and the feature extraction and prediction do not need to be explicitly defined [9].

In this paper, using the time-series throughput generated from a simulated factory scenario, we study the throughput

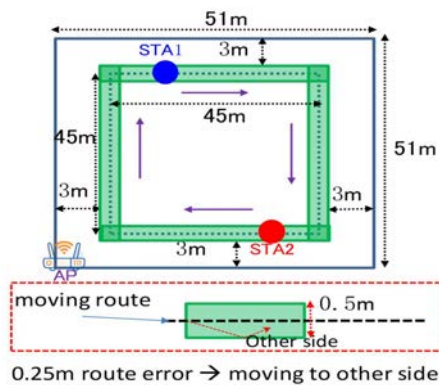


Fig. 1. Simulated factory environment.

prediction using CNN to show its prediction performance. Different with image or numerical recognition using CNN, in which the distribution of the target value during training stage usually has a similar level, the distribution of the throughput data concentrates only on several values. This centralized distribution may degrade the prediction accuracy because the weights of the neural network are mainly decided by several values with major distribution as targets during the training stage. Therefore, we will propose to introduce vectorization of the prediction target. This can reduce the centralization of the distribution. The simulated results confirm that the propose method can improve the prediction accuracy.

II. THE SIMULATION SETUP FOR TIME-SERIES THROUGHPUT DATA AND EXPLANATION OF CONVOLUTIONAL NEURAL NETWORK (CNN)

A. Generation of time-series throughput data

To get the time-series throughput data for throughput prediction, we simulated a 51m-by-51m factory scenario with two moving stations (STAs). The simulated scenario is shown in Fig. 1. The two mobile STAs are assumed to be high-resolution monitoring cameras which capture the real-time data and send it to the access point (AP) side. The moving routes for both STAs are clockwise with a speed at 1 m/s. The distance between two STAs is about 90 m. The moving route includes route error within about ± 0.25 m. When moving STA deviates the main route to ± 0.25 m, the moving direction will be changed to other side. The simulation parameters are listed in Table I. We employ a ray-trace channel model [11] to get the channel variation and then RSS values. The simulation sets 10 constant scattered points to generate the reflected paths between the AP and each location of STA1 and STA2. The channel response is calculated using the distances of the reflected paths and the LOS path.

We run 8-hour simulation and the RSS of links between each STA and AP has been recorded for further simulation to get the throughput results. Each RSS value is obtained by averaging all RSS samples during one second. After obtaining the RSS values of two links, we calculate the SNR values of both links between AP and two STAs with the assumption that noise floor is about -90 dBm for a 20 MHz wireless LAN channel. Then the respective modulation and coding scheme

TABLE I
SIMULATION PARAMETERS FOR RAY-TRACING CHANNEL MODEL.

Item	Value
Location of AP	(0 m, 0 m)
Start point (STA1 STA2)	(3 m, 3 m), (48 m, 48 m)
Moving speed (STA1, STA2)	1 m/s, 1 m/s
Transmit power (STA1, STA2)	0 dBm, 0 dBm
Center frequency (GHz)	2.4
Moving route error	below 0.5 m/s
Number of scattered positions	10
Scattered positions X-axis (meter)	5, 11, 22, 25, 33, 37, 30, 42, 58, 59
Scattered positions Y-axis (meter)	25, 37, 34, 28, 51, 12, 45, 26, 35, 45
Path loss model (dB)	$18.7\log_{10}(d)+46.8+20\log_{10}(f/5 \text{ GHz})$
Noise floor (20MHz)	-90 dBm
RSS samples per second	32

TABLE II
THE RELATIONSHIP BETWEEN SNR RANGE AND DATA RATE.

SNR range [dB]	>40	35-40	30-35	25-30
Data rate [Mbps]	54	48	36	24
SNR range [dB]	20-25	15-20	10-15	<10
Data rate [Mbps]	18	12	9	6

(MCS) index or data rate is determined from the SNR value for each second with the relationship shown in Table II. After obtained the MCS regime, we can use a theoretical throughput model [12] to generate the theoretical throughput of the two links with considering different requirements of traffic for different applications.

To get the time-series throughput of each link, we assume that both STA1 and STA2 have 4 cameras which face different directions and each camera requires 510 kbps throughput at STA1 and 760 kbps throughput at STA2 for video transmission, respectively. Therefore, the total required link throughput between AP and STA1, STA2 are about 2 Mbps and 3 Mbps, respectively.

The throughput is calculated using the generated MCS value from the ray-trace model in each second. We have generated 8-hour theoretical throughput data and the 30-min theoretical throughput of both links are shown in Fig. 2. As shown in figure, the throughput of both links varies according with time. The reason is that the moving route error will make the RSS different even if the route is planned as 45 m each side. The different RSS changes the throughput between two links, and it becomes more unpredictable.

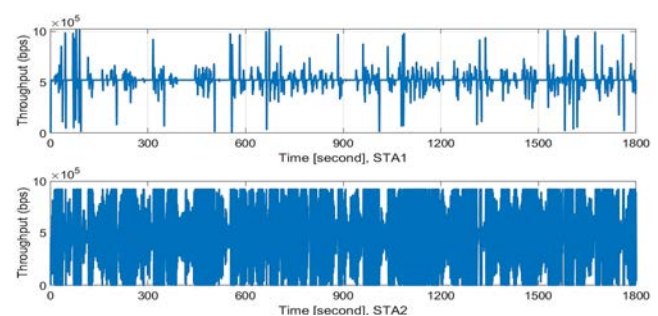


Fig. 2. The generated partial time-series throughput data of two STAs.

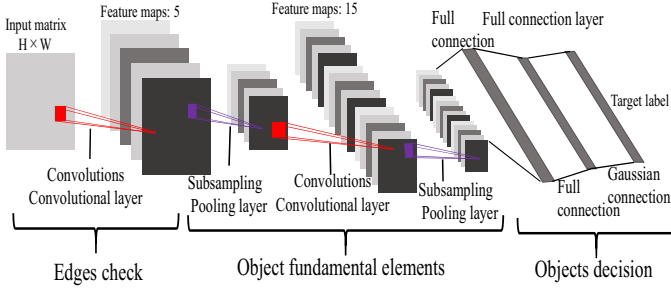


Fig. 3. The structure of CNN.

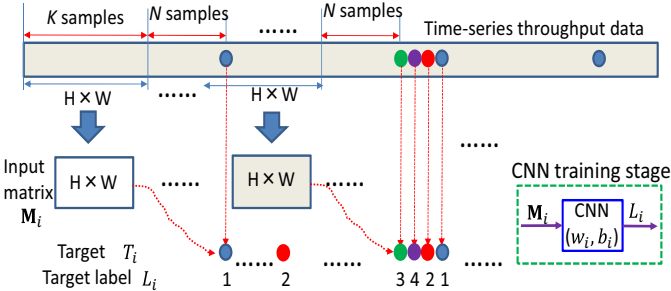


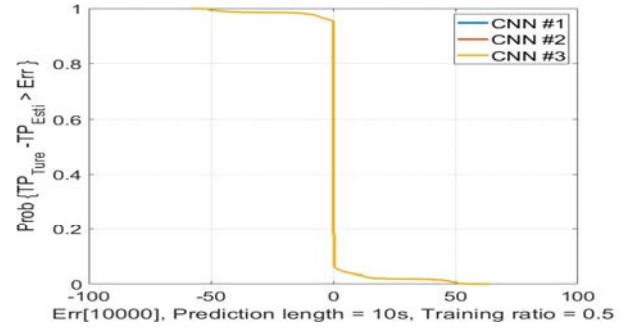
Fig. 4. The training stage of CNN using throughput data.

B. Convolutional neural network (CNN)

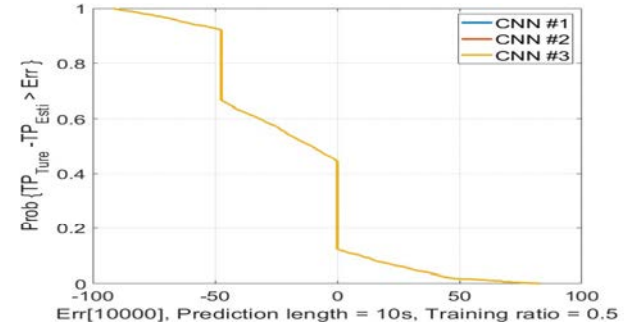
CNN is a deep learning algorithm and its architecture is analogous to the connectivity pattern of neurons in the human brain [14]. Therefore, CNN can be used for image identification, image classification, and can be utilized to detect various aspects/objects in the image to differentiate one from the others [15]. In addition, CNN can also be used for the prediction of time-series data [16]. Readers can find the related contents in [16].

Figure 3 shows the fundamental structure of CNN and the related functions. Generally, CNN includes three major layers: convolutional layer, pooling layer and fully-connected layer. The convolutional layer computes the output of neurons that are connected to local regions in the input matrix. All neurons compute a dot product between their weights and a small region through which they are connected to the input matrix. Such local regions can be defined as a filter with small width and height to reduce the computational complexity. The output of each filter is called a “feature map” which shows the effect of an edge check. Different filters generate a different feature map which shows the details of input matrix. Then the pooling layer performs a down-sampling operation along the spatial dimensions and places the data in a small volume. CNN can have several convolutional layers and pooling layers to dig the fundamental elements of image data. It should be noted that between the convolutional layer and the pooling layer, usually, there is a rectified linear unit (ReLU) layer which applies an element-wise activation function, such as the $\max(0, x)$ threshold at zero to remove some unimportant data x to simplify the representation of fundamental elements. Finally, the fully-connected (FC) layer combines all fundamental elements of image data to provide the final decision.

From the explanations of CNN structure, we can find that the data process of CNN is similar to the process in which



(a) Time-series throughput data of STA1



(b) Time-series throughput data of STA2

Fig. 5. Prediction performance of three CNNs ($H = W = 16$).

a human watches objects and then finds the different features (details) of the objects, and finally recognizes what the human watches based on these features. For the detail of CNN, readers can find contents in [15].

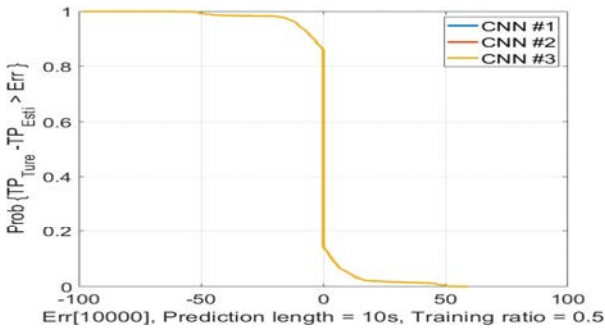
The training stage of time-series prediction using PNN is shown in Fig. 4. For CNN training, the i -th input matrix and its target are given as M_i and T_i , respectively. The prediction length is assumed to be N samples. For each input matrix with width W and height H , K samples are used to form an input matrix so that $K = H \times W$. Before starting the training, the target T_i is labeled as L_i which represents that the T_i belongs to the L_i -th group which includes real throughput data of training target. The training process is to find the optimal weight of CNN as (w_i, b_i) for each neuron to make the loss function as small as possible. After the training stage, for each new input matrix, the respective label is calculated using the weight of each neuron. Finally, from the decided label, the value of predicted throughput is determined.

III. THROUGHPUT PREDICTION OF PREDICTION METHOD USING CNN

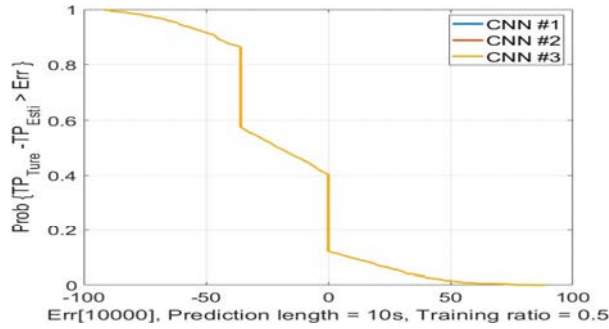
To evaluate the prediction performance, we first select the parameters for CNN. The CNN includes many parameters such as the numbers of feature maps (FM), the number of convolutional and pooling layers and fully-connected layers. When an input matrix includes many details to learn, it is better to set a large number of feature maps and more layers of the convolutional, pooling and be fully-connected. In addition, the size of filters can be set as small as possible to capture more details. In this paper, the input matrix is formed from the time-series throughput data which has limited information

TABLE III
THE PARAMETERS OF THREE CNNs USED FOR THROUGHPUT PREDICTION.

	CONV+Pooling Layers				FC Layers	
CNN#1	CONV		Pooling		Node: Num. of labels Activa. Func.: ReLU	
	Filter: 13-by-13 Feature Map: 1 Activa. Func.: ReLU		Mat. Size: 2-by-2 Subsamp. Med.: mean Activa. Func.: ReLU			
CNN#2	CONV		Pooling		Node: Num. of labels Activa. Func.: ReLU	
	Filter: 7-by-7 Feature Map: 10 Activa. Func.: ReLU		Mat. Size: 2-by-2 Subsamp. Med.: mean Activa. Func.: ReLU			
CNN#3	CONV	Pooling	CONV	Pooling	FC1	FC2
	Filter: 7-by-7 Feature Map: 10 Activa. Func.: ReLU	Mat. Size: 2-by-2 Subsamp. Med.: mean Activa. Func.: ReLU	Filter: 13-by-13 Feature Map: 40 Activa. Func.: ReLU	Mat. size: 2-by-2 Subsamp.Med.: mean Activa. Func.: ReLU	Node: 100	Node: Num. of labels Activa. Func.: ReLU



(a) Time-series throughput data of STA1



(b) Time-series throughput data of STA2

Fig. 6. Prediction performance of three CNNs ($H = W = 34$).

for learning. Therefore, we use two simple CNNs to predict the throughput, and another complicated CNN is also used for comparison. The parameters of three CNNs are listed in Table III. Here CNN#1 is the simplest one which has only one convolutional, pooling and FC layers. In addition, it has only one feature map with a filter size of 13×13 . Compared with CNN#1, CNN#2 increases the number of feature maps and filter size to 10 and 7×7 which tries to capture more details of input data. Finally, CNN#3 is a more complicated one which has two sets of convolutional, pooling and FC layers and the number of feature maps is increased to 40. For prediction length, we set the value of N as 10 samples.

In the training process, all CNNs utilize a forward phase and backward phase as training weight (W_i, b_i) to make the cross-entropy loss as small as possible. The major idea for the training process is based on gradient descent in an iterative

way. Readers can find its detail in [16].

From Fig. 1, we can find that each STA can generate 3-minute data after each round. The 8-hour data is separated into training data and validation data. All throughput data generated at the odd round is used as training data, and the throughput data generated at the even round is used as validation data. The complementary cumulative distribution function (CCDF) of the prediction error of throughput data using three CNNs are shown in Fig. 5 with ($H = W = 16$) and Fig. 6 with ($H = W = 34$), respectively. The training process has been operated over 400 times and makes the cross-entropy loss of all CNNs smaller than 0.1. All figures show that all the three CNNs have almost identical prediction accuracy. The results imply that the throughput data from both STAs are all simple. Therefore, the simplest structure of CNN (CNN#1) is enough to fully capture the feature of time-series throughput data. On the other hand, CNNs with smaller input matrix size have better prediction accuracy than that of the larger input matrix size (in Fig. 6). The results imply that using a small size of input matrix can further capture the data pattern for CNNs.

IV. PREDICTION PERFORMANCE IMPROVEMENT USING TARGET VECTORIZATION FOR CNN

CNN was firstly used for the recognition from handwriting image which can achieve over 93% accuracy [17]. After that, CNN has been widely utilized for many different areas. The CNN can obtain promising performance especially when the targets of input data or image have an even distribution. When target includes a few values which occupy major distribution, the prediction process is apt to make the predicted value near to these values after training process. Therefore, some rare values are neglected which may reduce the prediction accuracy.

One efficient way to reduce such prediction loss is to make vectorization of target. Figure 7 shows the idea of target vectorization. Suppose there is an on-off time-series data, and it only has two values as on (1) with 70% and off (0) with 30%, respectively. For the i -th input matrix M_i of the conventional CNN, the target is usually a scalar (0 or 1) as T_i . Therefore, as shown in Fig. 7, the training process finds the weight to make the predicted results as 1 or 0 with 70% or

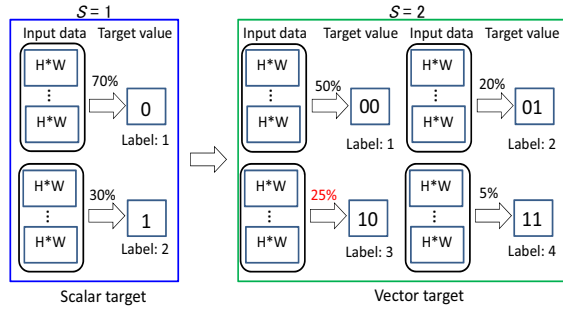


Fig. 7. The main idea of target vectorization for CNN.

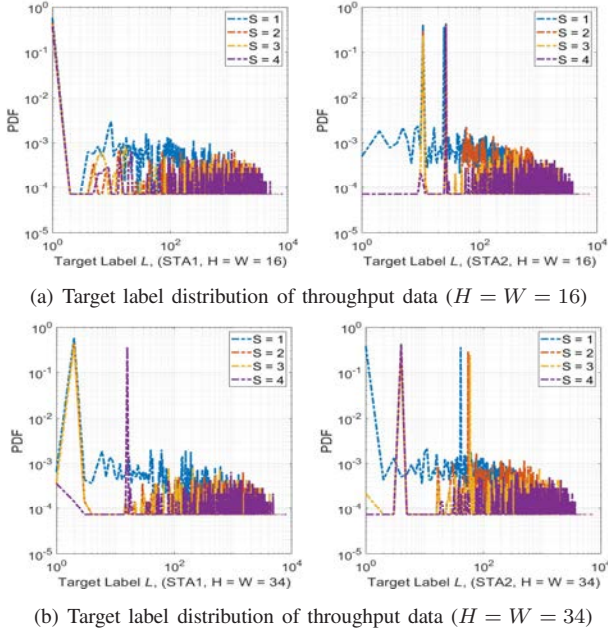
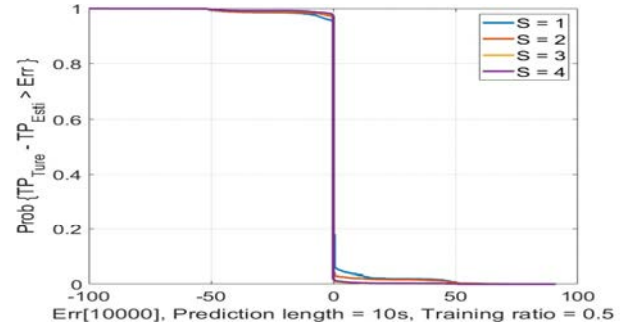


Fig. 8. Target label distribution of throughput data at both STAs.

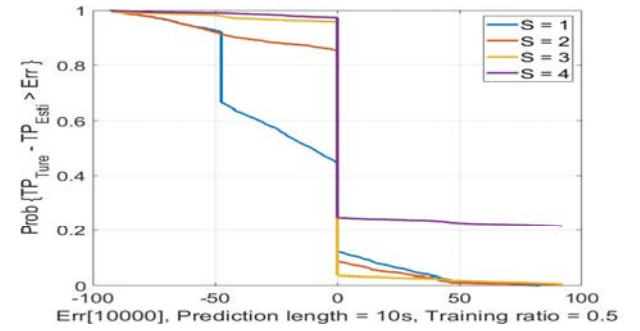
30%, respectively. Therefore, even if some rare input matrices appear, the predicted values are also judged as 1 or 0 with 70% or 30%, respectively.

Different with a single target ($S = 1$), target vectorization can make the target value is not a scalar but a vector as $[T_i, T_{i+1}, \dots, T_{i+S-1}]$ with label L_i for the i -th input matrix M_i . With this simple process, it can improve the prediction accuracy for the occurrence of some rare input matrices. As an example with ($S = 2$) in Fig. 7, two consecutive data are used as one target for each input matrix. In such case, the target vector can be four candidates with different probabilities. This process makes the distribution more even than that of ($S = 1$).

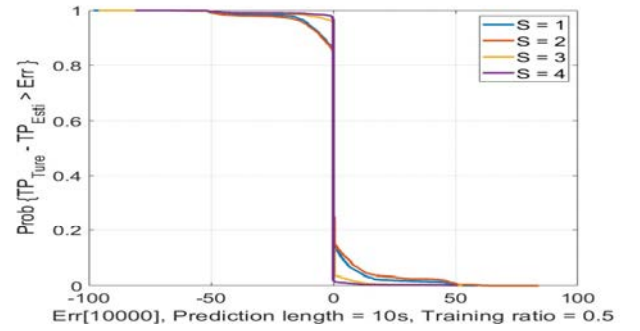
Figure 8 shows the target label (L) distribution of throughput data of both STAs for different sizes of the input matrix. Here the X-axis is label index and Y-axis is the distribution of the label index. The size of target vector is changed from $S = 1$ to $S = 4$. It is shown that the distribution of label after target vectorization is spread over a wider range as S is increased. When $S = 1$, the throughput data is mainly concentrated on the first label for throughput data at STA1. For the throughput data at STA2, two labels are dominant. The distribution property of STA2 seems to have an impact on the prediction accuracy because CCDF values also have two



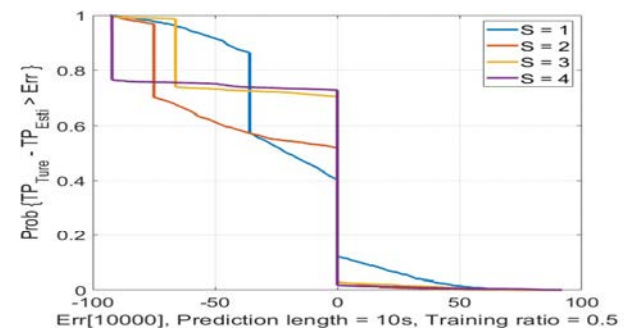
(a) Time-series throughput data of STA1



(b) Time-series throughput data of STA2

Fig. 9. Prediction performance of CNN#1 with different S ($H = W = 16$).

(a) Time-series throughput data of STA1



(b) Time-series throughput data of STA2

Fig. 10. Prediction performance of CNN#1 with different S ($H = W = 34$).

clusters as shown in Fig. 5(b) and Fig. 6(b). When $S = 4$, more labels appears which makes the CNN have more options for its input. Therefore, it benefits the improvement of prediction accuracy.

Figures 9 and 10 show the prediction performance of CNN#1 using the proposed target vectorization where the sizes

of input matrix are ($H = W = 16$) and ($H = W = 34$), respectively. The parameters of CNN#1 and calculation process are the same with the previous section. The prediction accuracy is improved for both the throughput data from STA1 and STA2. For the throughput data from STA1, the improvement of prediction accuracy from $S = 1$ to $S = 4$ is about 8% for the case of ($H = W = 16$) and 28% for the case of ($H = W = 34$), respectively. On the other hand, for throughput data from STA2, CNN#1 using the proposed method with $S = 3$ and the size of input matrix as ($H = W = 16$) can dramatically improve its prediction accuracy than that of the original method using $S = 1$.

V. CONCLUSION

This paper studied the throughput prediction with CNN for the time-series throughput data generated from a simulated factory scenario. Different with image or numerical recognition using CNN, the distribution of targets of time-series throughput data concentrates only on several values. This centralized distribution may degrade the prediction accuracy. This paper proposed one method by increasing the size of target value to reduce the centralization of the distribution. Then, it was confirmed that the proposed method makes the options of the output having more variety and improves the prediction accuracy of CNN.

ACKNOWLEDGMENT

This work includes results of the project entitled “R&D on Technologies to Densely and Efficiently Utilize Radio Resources of Unlicensed Bands in Dedicated Areas,” which is supported by the Ministry of Internal Affairs and Communications as part of the research program “R&D for Expansion of Radio Wave Resources (JPJ000254)”.

REFERENCES

- [1] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Fu and O. Postolache, “Internet of things for smart ports: Technologies and challenges,” *IEEE Instrumentation & Measurement Magazine*, vol. 21, no. 1, pp. 34–43, February 2018.
- [2] E. A. Oyekanlu et al., “A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications,” *IEEE Access*, Early Access.
- [3] Z. M. Fadlullah et al., “State-of-the-art deep learning: Evolving machine intelligence toward tomorrow’s intelligent network traffic control systems,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, Fourth-quarter 2017.
- [4] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, “A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1790–1821, third-quarter 2017.
- [5] M. Mirza, J. Sommers, P. Barford, and X. Zhu, “A machine learning approach to TCP throughput prediction,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1026–1039, Aug. 2010.
- [6] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei, “LinkForecast: cellular link bandwidth prediction in LTE networks,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1582–1594, July 2018.
- [7] A. Biernacki, “Improving quality of adaptive video by traffic prediction with (F)ARIMA models,” *Journal of Communications and Networks*, vol. 19, no. 5, pp. 521–530, Oct. 2017.
- [8] J. Webber, N. Suga, A. Mehbodniya, K. Yano, and T. Kumagai, “Study on fading prediction for automated guided vehicle using probabilistic neural network,” in Proc. 2018 Asia-Pacific Microwave Conf. (APMC), Nov. 2018.
- [9] M. Cheung, J. Shi, O. Wright, L. Y. Jiang, X. Liu and J. M. F. Moura, “Graph signal processing and deep learning: Convolution, pooling, and topology,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 139–149, Nov. 2020.
- [10] M. N. Bajwa et al., “Confident classification using a hybrid between deterministic and probabilistic convolutional neural networks,” *IEEE Access*, vol. 8, pp. 115476–115485, 2020.
- [11] C.-F. Yang B.-C. Wu C.-J. Ko, “A ray-tracing method for modeling indoor wave propagation and penetration,” *IEEE Trans. Antennas Propag.*, vol. 46 no. 6 pp. 907–919 Jun. 1998.
- [12] T. Iwami, I. Roslan, Y. Takaki, K. Yamori, C. Ohta and H. Tamaki, “Utility based distributed association control scheme with user guidance for IEEE802.11 wireless LANs,” *IEICE Transactions on Communications*, Vol. E98-B, No. 8, pp. 1700–1714, Aug. 2015.
- [13] N. Suga, K. Yano, J. Webber, Y. Hou, T. Higashimori and Y. Suzuki, “Prediction of QoS outage probability for wireless communication in factory environments,” in Proc. 2019 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Tunis, Tunisia, pp. 124–129, 2019.
- [14] C. Du, C. Du, L. Huang and H. He, “Reconstructing perceived images from human brain activities with bayesian deep multiview learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2310–2323, Aug. 2019.
- [15] S. Ji, W. Xu, M. Yang and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
- [16] C. Liu, W. Hsiao and Y. Tu, “Time series classification with multivariate convolutional neural network,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4788–4797, June 2019.
- [17] A. Mahmood et al., “Capacity and frequency optimization of wireless backhaul network using traffic forecasting,” *IEEE Access*, vol. 8, pp. 23264–23276, 2020.

Yafei Hou received his Ph.D. degrees from Fudan University, China and Kochi University of Technology (KUT), Japan in 2007. He is assistant Professor at the Graduate School of Natural Science and Technology, Okayama University, Japan. He is also a guest research scientist at Wave Engineering Laboratories, ATR Institute International, Japan.

Kazuto Yano received the B.E. degree in electrical and electronic engineering, and the M.S. and Ph.D. degrees in communications and computer engineering from Kyoto University in 2000, 2002, and 2005, respectively. Currently, he is Head of Dept. Wireless Communication Systems at Wave Engineering Laboratories, ATR Institute International, Japan.

Norisato Suga is assistant professor at faculty of engineering, Tokyo University of Science, Japan and Tokyo, Japan and guest research scientist at Wave Engineering Laboratories, ATR Institute International, Japan.

Webber Julian is assistant professor at Graduate School of Engineering Science, Osaka University, Japan and guest research scientist at Wave Engineering Laboratories, ATR Institute International, Japan.

Eiji Nii is a researcher scientist at Wave Engineering Laboratories, ATR Institute International, Japan

Toshihide Higashimori received the B.E. degree in economics from Kindai University, Japan, in 1989. He joined the Wave Engineering Laboratories, ATR Institute International, Japan and is engaging in development of several simulation softwares.

Satoshi Denno received the M.E. and Ph.D degrees from Kyoto University, Kyoto, Japan in 1988 and 2000, respectively. He is a full professor at graduate school of natural science and technology, Okayama University.

Yoshinori Suzuki received the B.E., M.E. and Ph.D. degrees from Tohoku University, Sendai, in 1993, 1995 and 2005 respectively. He is the director of ATR Wave Engineering Laboratories, Kyoto, Japan.