DeepSIP: A System for Predicting Service Impact of Network Failure by Temporal Multimodal CNN

Yoichi Matsuo*, Tatsuaki Kimura†, Ken Nishimatsu*
* NTT Network Technology Laboratories, NTT Corporation
Email:{yoichi.matsuo.ex, ken.nishimatsu.hd}@hco.ntt.co.jp
† Department of Information and Communications Technology, Graduate School of Engineering, Osaka University
Email:kimura@comm.eng.osaka-u.ac.jp

***Abstract*—When a failure occurs in a network, network operators need to recognize service impact, since service impact is essential information for handling failures. In this paper, we propose Deep learning based Service Impact Prediction (DeepSIP), a system to predict the time to recovery from the failure and the loss of traffic volume due to the failure in a network element using a temporal multimodal convolutional neural network (CNN). Since the time to recovery is useful information for a service level agreement (SLA) and the loss of traffic volume is directly related to the severity of the failures, we regard these as the service impact. The service impact is challenging to predict, since a network element does not explicitly contain any information about the service impact. Thus, we aim to predict the service impact from syslog messages and traffic volume by extracting hidden information about failures. To extract useful features for prediction from syslog messages and traffic volume which are multimodal and strongly correlated, and have temporal dependencies, we use temporal multimodal CNN. We experimentally evaluated DeepSIP and DeepSIP reduced prediction error by approximately 50% in comparison with other NN-based methods with a synthetic dataset.**

***Index Terms*—Service impact, Network operation, Multimodal, Time series, CNN**

I. Introduction

Service impact, such as time to recovery due to a failure in a network and loss of traffic volume caused by a failure, needs to be predicted by network operators. This is because the available time of network service is guaranteed as a service level agreement (SLA) between network providers and customers. Additionally, if network operators can recognize the service impact before recovering the network, they can prioritize the order of recovery. For example, if a failure has no or little impact on a service, network operators do not have to conduct repairs immediately. Therefore, the prioritization will contribute to reducing operating expenses.

Even if a network failure is caused by a single network element failing, the service impact is challenging to predict for three reasons. First, a network element does not explicitly contain any information about service impact due to a failure. Second, the temporal characteristics of service impact are highly diverse depending on what type of network failure occurs. For instance, a failed module of routers generates more service impact than link flapping since it takes more time to replace a module than to recover from link flapping by quickly rebooting. Third, even if the same type of network failure occurs, the service impact is not always the same. This is because the service impact also depends on both traffic volume at the failure and past traffic volume.

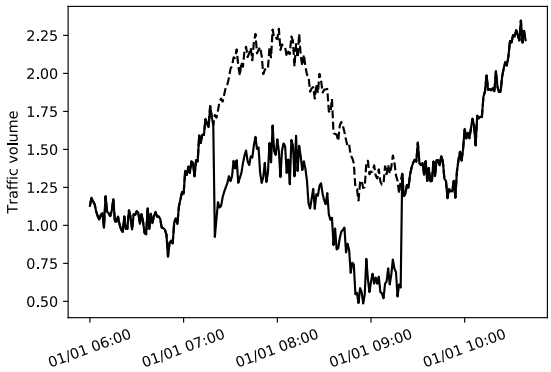


Fig. 1: Example of a service impact

We propose DeepSIP[1], a system to automatically predict the time to recovery from the failure and the loss of traffic volume due to the failure in a network element from syslog messages and traffic volume using temporal multimodal convolutional neural network (CNN). Since the loss of traffic volume is directly related to the severity of the failures, and represents the impact of the failures on customers, we regard the loss of traffic volume and time to recovery as types of service impact. Figure 1 is an example of what DeepSIP predicts. In this figure, a failure occurred at around 7:20 and recovery finished at around 9:20. The solid line is actual traffic volume and the dashed line is traffic volume if the failure had not occurred. DeepSIP predicts the time to recovery which is 120 min, and the loss of traffic volume, which is the area between the solid and dashed lines in this example.

The time to recovery and the loss of traffic volume are challenging to predict since the network element do not explicitly contain information about service impact. However, we consider that syslog messages and traffic volume may contain such information. Since syslog messages describe what occurs in the network element, they contain the hidden information

978-1-7281-4973-8/20/$31.00 © 2020 IEEE

[1]DeepSIP means **Deep** learning based **S**ervice **I**mpact **P**rediction system

of what types of failures occur in network elements. Since the time to recovery depends on recovery procedures, the types of failures are useful information. For example, link flapping can be recovered by rebooting the interface, whereas a module fault can be recovered by replacing the module. The loss of traffic volume depends on how much traffic volume is reduced due to the failure and the time to recovery. Thus, we argue that the time to recovery and the loss of traffic can be predicted by information of a failure type inferred from syslog messages and traffic volume at the time before a failure occurs.

To extract useful information for prediction from syslog messages and traffic volume are also challenging. First, network data are multimodal, which means there are continuous values such as traffic volume and text data such as syslog messages. We use a CNN to separately extract useful features using preprocessed vectors and scalar values generated from syslog messages and traffic volumes separately. Second, syslog messages and traffic volume are strongly correlated. Thus, we use another CNN to extract the correlation between syslog messages and traffic volume. Finally, syslog messages and traffic volume have long and short temporal dependencies. Therefore, we again use a CNN for time-series data that can extract these dependencies automatically. We evaluate the effectiveness and the accuracy of DeepSIP by comparing it with other NN-based methods.

The rest of this paper is organized as follows. We describe related works in Section II. In Section III, we explain DeepSIP, a system to predict service impact. We evaluate DeepSIP in Section IV. Finally we conclude this paper in Section V.

II. Related Work

A. Network Operation

Many studies about network operations such as anomaly detection, root cause analysis and recovery from the failures have been developed [1]–[3]. PRISM [1] is a system for detecting performance change of network elements due to maintenance by singular value decomposition. However, only a study [4] have focused on service impact despite its importance. Yang et al. [4] considered the number of affected user devices as service impact and modeled how many user devices are affected in a cellular environment when a base station fail. Unlike their analysis [4], our system takes into account the difference in types of failures to predict the time to recovery and the loss of traffic volume.

B. Deep Learning Algorithm for Time-series Data

Deep learning based regression methods have been proposed in recent years. Representative methods for time-series data, are a long short-term memory (LSTM) [5]. However, it has been reported that LSTM cannot extract long-time temporal dependencies in the data [6]. Quasi-recurrent NN (QRNN) [6] proposed to deal with long time dependencies. Applying a CNN to time-series data enables us to directly input whole time-series data. Traffic volume prediction method using deep learning for time-series data have also been developed [7]. Wang et al. [8] proposed a combination of an autoencoder and LSTM model to predict traffic volumes of each link in the network.

However, the above methods focus only on unimodal such as audio data and traffic data. Regarding multimodal data, DeepSense [9] was proposed for classification or regression tasks using time-series sensor data. It combines a CNN and gated recurrent unit (GRU), which is an LSTM-based method. However, since LSTM cannot extract long-time temporal dependencies, we propose applying QRNN instead, to extract long-time temporal dependencies from the multimodal time-series data.

TABLE I: Definition of symbols

Symbol	Definition
t	Time slot
T	Time slot in which a failure occurred
TTR	Time slot in which recovery finished
V_{TTR}	Loss of traffic volume between T and TTR
TTR', $\hat{V}_{\mathsf{TTR}'}$	Actual data in the past
$\{\cdot_t\}_{t=1}^{T}$	Set of time-series data
$\boldsymbol{x}_t$	Vector at time t that represents preprocessed syslog messages
$\{\boldsymbol{x}_t\}_{t=1}^{T}$	Time-series data of $\boldsymbol{x}_t$
y_t	Scalar value at time t that represents traffic volume
$\mathcal{D}$	training dataset, $\{(\{\boldsymbol{x}_t\}_{t=1}^{T}, \{y_t\}_{t=1}^{T}, \mathsf{TTR}', \hat{V}_{T\to\mathsf{TTR}'})_d\}_{d=1}^{\lvert\mathcal{D}\rvert}$
$\{y_t\}_{t=1}^{T}$	Time-series data of y_t
w_i	Weight of CNN in layer i
$z^{(i)}$	Output of CNN from layer i

III. DeepSIP

In this section, we explain our system, DeepSIP. In short, when a network element fails, DeepSIP predicts the service impact of the failure, i.e., the time to recovery from the failure and the loss of traffic volume due to the failure, using past syslog messages and traffic volume of the network element. The loss of traffic volume is directly related to the severity of the failures and represents the impacts of the failures on customers using the network at that time. Moreover, the approximate number of affected customers can be estimated using the average loss of traffic volume which is calculated by the time to recovery, and average sending or receiving traffic volume per customer. Therefore, DeepSIP predicts the time to recovery and the loss of traffic volumes.

Since syslog messages and traffic volume are time-series data with different features, which means that there are continuous values such as traffic volumes and text data such as syslog messages, DeepSIP uses the CNN for multimodal data. Moreover, we propose applying QRNN instead of LSTM, to extract long-time temporal dependencies from the multimodal time-series data.

In the following subsections, we first define the considered problem. Then, we give an overview of DeepSIP. Finally, we give details of the temporal multimodal CNN in DeepSIP. Notations used in this paper are summarized in Table I.

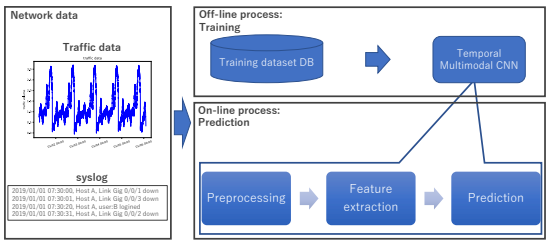


Fig. 2: System Overview of DeepSIP

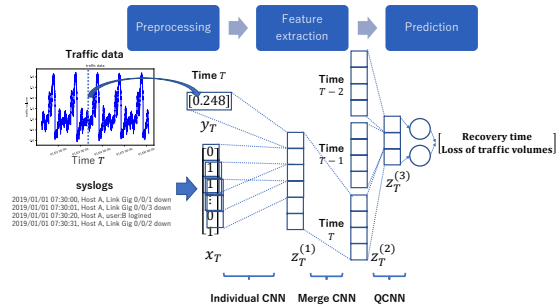


Fig. 3: Temporal multimodal CNN in DeepSIP

A. Service Impact Prediction Problem

We explain the problem formulation of predicting service impact. Let T be the time slot in which a failure occurred and TTR be the time slot in which recovery finished, where time t is slotted as $t = 1, 2, \ldots, T, \ldots, \mathsf{TTR}$. Let V_{TTR} be the scalar value that represents loss of traffic volume between T and TTR due to a failure. We denote a set of time-series data as $\{\cdot_t\}_{t=1}^{T}$. Let $\boldsymbol{x}_t$ be a vector at time t and $\{\boldsymbol{x}_t\}_{t=1}^{T}$ be time-series data of $\boldsymbol{x}_t$. Similarly, let y_t be a scalar value at time t and $\{y_t\}_{t=1}^{T}$ be a time-series data of y_t. These vectors and scalar values are generated from syslog messages and traffic volume respectively, and input into the temporal multimodal CNN. We explain how to generate these vectors and scalar values in Section III-C. The goal of DeepSIP is to predict the TTR and V_{TTR} from the collected syslog messages and traffic volume in the past and at the time when the failure occurs. Even though DeepSIP uses traffic volume at T as input, the loss of traffic volume V_{TTR} is difficult to predict since it depends on the type of failure. For instance, we cannot distinguish the difference in traffic changes in a network element between spike and packet drops when a failure has just occurred. DeepSIP predicts them using syslog messages that contain the types of failures in network element.

B. System Overview

We show the system overview of DeepSIP in Figure 2. DeepSIP consists of two phases: a training phase, which is an off-line process, and a predicting phase, which is an on-line process. In the training phase, weights in DeepSIP are trained using training dataset $\mathcal{D}$. Superscript $'$ denotes actual data from past failures in the training dataset. Let TTR' be a time to recovery at the past failure and $V'_{\mathsf{TTR}'}$ be a loss of traffic volume at the past failure. Training data for DeepSIP consist of a tuple, $(\{\boldsymbol{x}_t\}_{t=1}^{T}, \{y_t\}_{t=1}^{T}, \mathsf{TTR}', V'_{\mathsf{TTR}'})$, and the $\mathcal{D}$ is a set of tuples, $\mathcal{D} = \{(\{\boldsymbol{x}_t\}_{t=1}^{T}, \{y_t\}_{t=1}^{T}, \mathsf{TTR}', V'_{T\to\mathsf{TTR}'})_d\}_{d=1}^{|\mathcal{D}|}$. Here $|\mathcal{D}|$ is the amount of data for training. In the predicting phase, DeepSIP predicts TTR and V_{TTR} from the collected syslog messages and traffic volume.

C. Temporal Multimodal CNN in DeepSIP

Temporal multimodal CNN in DeepSIP consists of three parts: preprocessing, feature extraction, and prediction. The temporal multimodal CNN in DeepSIP is illustrated in Figure 3. In the preprocessing part, DeepSIP generates preprocessed vectors $\{\boldsymbol{x}_t\}_{t=1}^{T}$ and scalar values $\{y_t\}_{t=1}^{T}$ from syslog messages and traffic volume. In the feature extraction part, DeepSIP extracts the features of $\boldsymbol{x}_t$ and y_t, correlations between $\boldsymbol{x}_t$ and y_t, and temporal dependencies of $\{\boldsymbol{x}_t\}_{t=1}^{T}, \{y_t\}_{t=1}^{T}$ using CNNs.

1) Preprocessing: In the preprocessing part, we first split syslog messages and traffic volume accordance with a time slot. Then, to be able to handle syslog messages as numerical values, we use *template methods* [3], [10]. These methods are for automatically assigning a template ID to each message of the syslog. If syslog messages have the same meaning except the numerical part, such as timestamps or IP addresses, they are given the same template ID even if the messages of syslog differ. Examples of the templates are shown in Tab. II.

M is the size of $\boldsymbol{x}_t$ and is the total number of template IDs. Each element $x_{t,m}$ of $\boldsymbol{x}_t$ represents the number of occurrences of the m-th template ID within a time slot, τ. Similarly, y_t represents the traffic volume at time t. $\{\boldsymbol{x}_t\}_{t=1}^{T}$ and $\{y_t\}_{t=1}^{T}$ are the preprocessed vectors and scalar values, respectively, which are input for the temporal multimodal CNN.

2) Feature Extraction: In the feature extraction part, DeepSIP extracts features of syslog messages and traffic volume separately using CNNs. DeepSIP also extracts correlations between syslog messages and traffic volume, and time dependencies using CNNs. First, DeepSIP extracts internal features from multimodal data which are a preprocessed vector $\boldsymbol{x}_t$ and scalar value y_t at time t separately. We use 1d CNN, a CNN for vectors, for this extraction. The output vector $\boldsymbol{z}_t^{(1,\mathrm{syslog})}$ and its j-th element $z_{t,j}^{(1,\mathrm{syslog})}$ can be calculated by the following equations.

$$\boldsymbol{z}_t^{(1,\mathrm{syslog})} = \mathrm{1dCNN}^{(\mathrm{syslog})}(\boldsymbol{x}_t), \tag{1}$$

$$z_{t,j}^{(1,\mathrm{syslog})} = \sum_{k=0}^{K} w_k^{(\mathrm{syslog})} \times x_{t,k+j}, \tag{2}$$

where K is kernel size i.e., how many elements CNN convolutes, and $w_k^{(\mathrm{syslog})}$ are the weights of the CNN that are trained. The size of output vector $\boldsymbol{z}_t^{(1,\mathrm{syslog})}$ is $M-K$. Similar to $\boldsymbol{z}_t^{(1,\mathrm{syslog})}$, $\boldsymbol{z}_t^{(1,\mathrm{traffic})}$ is also calculated by 1d CNN.

$$\boldsymbol{z}_t^{(1,\mathrm{traffic})} = \mathrm{1dCNN}^{(\mathrm{traffic})}(y_t), \tag{3}$$

TABLE II: Examples of the templates of syslog messages

syslog messages	template ID	syslog template
2019/01/01 07:30:00, Host A, Interface Gig 0/0/1 down	1	XXX, Host A, Interface Gig XXX down
2019/01/01 07:30:10, Host A, Interface Gig 0/0/1 state changed	2	XXX, Host A, Interface Gig XXX state changed
2019/01/01 07:30:20, Host A, user:B logged in	3	XXX, Host A, XXX logged in
2019/01/01 07:30:31, Host A, Interface Gig 0/0/2 down	1	XXX, Host A, Interface Gig XXX down

where we set the kernel size as 1. We denote this layer of CNN as an individual the CNN following DeepSense.

Second, DeepSIP extracts the correlation between syslog and traffic volumes at time t. We arrange extracted features $\boldsymbol{z}_t^{(1,\mathrm{syslog})}$ and $\boldsymbol{z}_t^{(1,\mathrm{traffic})}$ into a vector $\boldsymbol{z}_t^{(1)}$ with size $M - K + 1$. We again use 1d CNN for this extraction.

$$\boldsymbol{z}_t^{(2)} = \mathrm{1dCNN}(\boldsymbol{z}_t^{(1)}), \quad z_{t,j}^{(2)} = \sum_{k=0}^{K} w_k^{(2)} \times z_{t,k+j}^{(1)}, \tag{4}$$

where $w_k^{(2)}$ are the weights of the CNN. We also denote this layer of the CNN as merge CNN following DeepSense.

Finally, DeepSIP extracts time dependencies of $\boldsymbol{z}_t^{(2)}$ between time series from 1 to T using QRNN. We arrange $\{\boldsymbol{z}_t^{(2)}\}_{t=1}^{T}$ in the column direction to create a matrix $Z^{(2)} \in \mathbb{R}^{N \times T}$, where N is the size of $\boldsymbol{z}_t^{(2)}$. DeepSIP outputs $Z^{(3)}$ using QRNN.

$$Z^{(3)} = \mathrm{QRNN}(\boldsymbol{z}_1^{(2)}, \ldots, \boldsymbol{z}_T^{2}), \quad z_{t,j}^{(3)} = \sum_{t=1}^{T} w_t^{(3)} \times \boldsymbol{z}_{t,t+j}^{(2)}, \tag{5}$$

where $w_t^{(3)}$ are weights of QRNN.

3) Prediction: In the prediction part, DeepSIP predicts time to recovery TTR and the loss of traffic volume V_{TTR} using $Z^{(3)}$ which has temporal and multimodal features in the syslog messages and traffic volumes. Predicting TTR and V_{TTR} is a regression task. This is done by linear functions, F_{TTR} and F_V, as in the following equations.

$$\mathsf{TTR} = F_{\mathsf{TTR}}(\boldsymbol{Z}^{(3)}), \quad V_{\mathsf{TTR}} = F_V(\boldsymbol{Z}^{(3)}), \tag{6}$$

When we train the model, we use the mean squared error as the loss function and update every weight of DeepSIP using SDG.

Note that the architecture of DeepSIP can be used for general purposes by changing the function in the prediction part. For classification, such as predicting the type of the failure, we use a softmax function to normalize the sum of the $\boldsymbol{Z}^{(3)}$ values to 1. By using this architecture, it is possible to extract the features of dependencies in the time series, those of each data and those between data for multimodal data.

IV. Evaluation

In this section, we evaluate of DeepSIP by using synthetic data that imitate realistic traffic volume and syslog messages. To extensively evaluate the accuracy of estimated service impacts by DeepSIP, we comprehensively generated synthetic traffic volumes and syslog messages, and injected degradation to these synthetic data to imitate the behavior of traffic volumes and syslog messages when failures occur. For realistic failures and corresponding degradation of traffic volumes, we referred to the idea about degradation patterns considered by Mahimkar et al. [1]. Using these data, we compared DeepSIP using the temporal multimodal CNN with base-line methods.

In what follows, first, we explain degradation patterns of traffic volumes when failures occur in the network. Then, we explain how to generate synthetic traffic volumes and syslog messages, and inject degradation at the time a failure occurs. Finally, we show experiment results and evaluations.

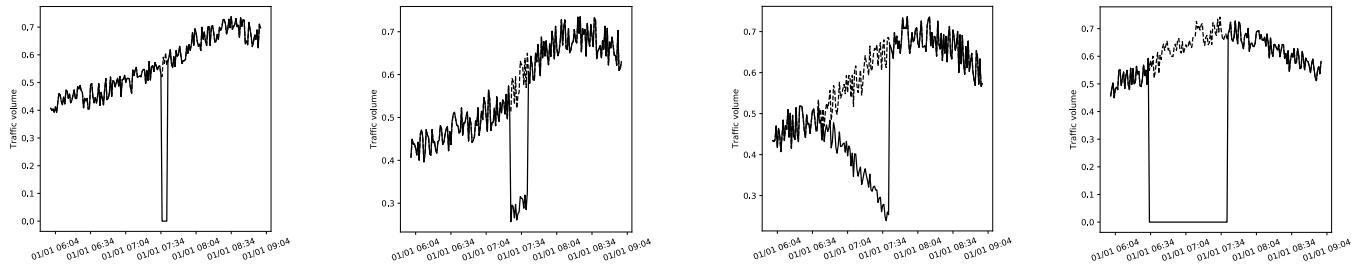
A. Failures and Corresponding Degradation Patterns

In this subsection, we explain traffic behaviors when failures occur. For evaluation, Mahimkar et al. [1] introduced realistic failures and corresponding degradation patterns of traffic volume, which are *spike*, *level-shift* and *ramp down*. We show examples of each degradation pattern of traffic volumes in Figures 4. The first pattern is spike in Figure 4 (a), which completely reduces traffic volume. An example of the failures that cause spike pattern is link flap. This is because link flap stops a network element from connecting to other network elements, but it is recovered quickly by rebooting the interface. The second pattern is level shift in Figure 4 (b), which reduces a certain amount of traffic volume during a certain period. An example of failures that cause the level-shift pattern is packet drops. This is because packet drops partially reduce traffic volume. The third pattern is ramp down in Figure 4 (c), which gradually reduces traffic volume. An example of the failures that cause ramp down pattern is deterioration of module in the network element. This is because deterioration of the module gradually progresses and the traffic in the deteriorated network element gradually reduces. In addition to the degradation patterns introduced by Mahimkar et al. [1], we added another degradation pattern of the traffic volumes. The fourth pattern is *long-period down* in Figure 4 (d), which completely reduces traffic volume. An example of the failures that cause long-period down is a hardware failure. This is because a hardware failure completely stop sending or receiving traffic.

B. Synthetic Traffic Volumes and Syslog

In this subsection, we explain how traffic volume and syslog messages in normal status were generated and then explain how degradation were injected into the data in normal status.

1) Synthetic Data in Normal Status: We created synthetic traffic volumes by superimposing sine waves with several frequencies. Typically, traffic volumes increase early in the



(a) Spike (b) Level shift (c) Ramp down (d) Long down

Fig. 4: Traffic pattern of failures

morning, decrease in the afternoon, and then increase in the evening and around night again. This is because office workers use the network when commuting in the morning and evening. Many people also use the network during their leisure time such as for watching videos after work. Therefore, we superimposed sine waves so that a superimposed wave has peak traffic in the evening and night. We also added white Gaussian noise to the superimposed wave.

We next explain how to generate synthetic syslog data. For synthetic syslog data, we did not generate syslog messages one by one, but generated vectors $\{\boldsymbol{x}_t\}_{t=1}^{T}$ described in Section III. To generate syslog in normal status, we categorize typical generation patterns of syslog into *periodic* and *random* events, and generated vectors on the basis of each generation pattern. A syslog message is periodically generated when a periodic event occurs such as a backup system running. Syslog messages are also randomly generated when the user event occurs such as a user log-in. To imitate periodic syslog generation, we prepared 10 template IDs for syslog which are generated by periodic events with every 2 min, 3 min, 5 min, 10 min, 30 min, 1 hour, 2 hours, 5 hours, 12 hours, and 24 hours, respectively. To imitate random syslog generation, we generated 30 other template IDs for syslog that are generated by random events. In this way, we generated vectors in normal status to imitate syslog messages in the network element. The generated syslog vectors correspond to approximately 1000 messages per day.

2) Synthetic Data in Abnormal Status: We next explain how to generate synthetic traffic volumes and syslog messages in abnormal status. First, we generated synthetic traffic volume at the time a failure occurred by injecting the degradation patterns into synthetic traffic volumes in normal status and set recovery time on the basis of recovery procedures described in Subsection IV-A. Then, we prepared syslog when a failure occurred by adding other template IDs to syslog vectors in normal status. When a failure occurs, syslog messages that are different from syslog messages in the normal status are typically generated. These syslog messages for each failure are also different from each other except those for ramp down. Therefore, we prepared 20 template IDs for each failure that represent syslog messages at time T. For ramp down, we randomly change generation patterns of periodic and random syslog messages. Although failures such as deterioration of the module do not completely stop systems, these failures make difficult for network operators to run back-up systems or log in to the system. Therefore, syslog messages might not be generated, but the frequencies of periodic syslog messages or random syslog messages in the network elements might be changed due to the failure.

The parameters of abnormal traffic patterns and syslog messages are summarized in Table III. We prepared synthetic traffic volumes and vectors in normal status for 6 days, and randomly injected abnormal behavior during the 6th day. The duration of a time slot is set as 1 min, With these settings, we prepared 9000 data for training and 1000 data for evaluation.

Since DeepSIP needs a number of data for training, we used the synthetic data for evaluation. However, we can also prepare a training dataset by injected the degradation patterns to real traffic data, as in the paper [1]. We can also apply DeepSIP trained in a test network environment to a real network environment by using a transfer learning approach [11]. A data augmentation technique [12] also enables us to increase the number of training dataset obtained from a real network. Therefore, proposed system is feasible and enables to apply to a real network operations.

C. Evaluation Results

Using the synthetic dataset, we evaluated the accuracy and effectiveness of DeepSIP. We implemented DeepSIP by Pytorch and used NVIDIA Tesla 100 GPU for training and evaluation.

We considered three different architectures as baseline: DeepSense, DeepSIP without merge CNN (D-SIP-w/o-Merge), DeepSIP without individual CNN (D-SIP-w/o-Indiv). Figures 5 and 6 show mean square errors and 95% confidence intervals. We set the prediction error as the relative error between the prediction and the ground truth. Figure 5 illustrates prediction results of V_{TTR}. For prediction of V_{TTR}, the results of DeepSIP were better than those of base-lines for every degradation pattern except long down pattern. For each pattern except long down pattern, DeepSIP reduced the prediction error by approximately 50% and the deviations of prediction

TABLE III: Summary of synthetic dataset

Traffic behavior	Example of failure	Time To Recovery	Traffic reduction	Number
Spike	Interface down	5min	Complete reduction	41-60
Level-shift	Packet loss	10min	50% Reduction	61-80
Ramp down	Module deterioration	60min	1% Reduction per 1 min.	1-40
Long-periodic down	Hardware failure	120min	Complete reduction	81-100

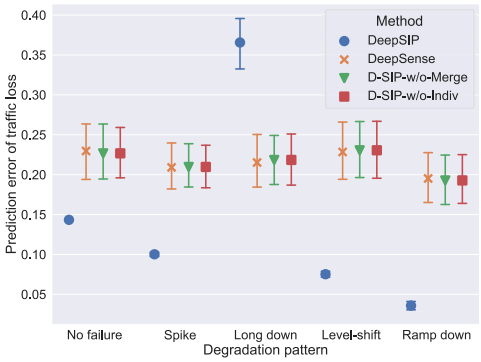


Fig. 5: Prediction error of V_{TTR}

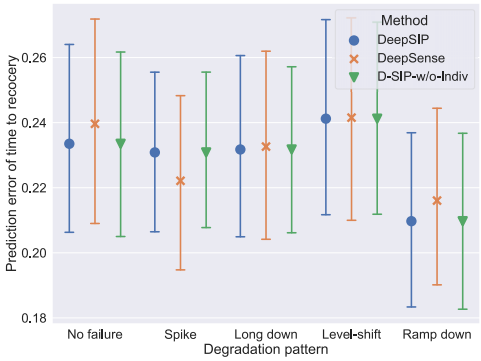


Fig. 6: Prediction error of TTR

errors of DeepSIP were quite smaller than those of other methods. This means that extracting the features using individual and merge CNN improved the prediction. DeepSIP reduced the prediction error comparing with DeepSense. This means that QRNN in DeepSIP also extracted temporal dependencies better than the GRU in DeepSense. Since the prediction errors of DeepSIP were quite small, this information is useful for network operators to prioritize failures and handle SLA.

Figure 6 illustrates prediction results of TTR. Since D-SIP-w/o-Merge did not converge to this problem, we eliminated the results of D-SIP-w/o-Merge from this Figure. This means that extracting the features using merge CNN improved the prediction. For prediction error of TTR, the results of DeepSIP were similar to the those of base-lines for every degradation patterns. For spike, long down and ramp down pattern, the results of DeepSIP were slightly better than those of base-lines. This means that individual CNN and QRNN in DeepSIP improved the predictions. The prediction errors of DeepSIP are 10 seconds for all degradation patterns. Thus this information is useful for network operators to prioritize failures and handle SLA.

V. Conclusion

Predicting the service impact is important for network operators since service impact is useful information for SLA and determining recovery procedure. In this paper, to analyze syslog messages and traffic volume which contain useful information for predicting the time to recovery and the loss of traffic volume, we used temporal and multimodal CNN. In the experiments, DeepSIP reduced prediction error by approximately 50% in comparison with base-lines. We showed the effectiveness of temporal multimodal CNN in DeepSIP by comparing with DeepSIP without individual CNN or merge CNN and DeepSense. Future works includes improving the prediction by customizing the loss function of DeepSIP and evaluating the effectiveness of DeepSIP with a real dataset.

References

[1] A. Mahimkar, Z. Ge, J. Wang, J. Yates, Y. Zhang, J. Emmons, B. Huntley, and M. Stockert, "Rapid Detection of Maintenance Induced Changes in Service Performance," *Proc. ACM CoNEXT*, 2011.

[2] Y. Matsuo, Y. Nakano, A. Watanabe, K. Watanabe, K. Ishibashi, and R. Kawahara, "Root-cause diagnosis for rare failures using bayesian network with dynamic modification," *Proc. ICC*, 2018.

[3] D. Min, L. Feifei, Z. Guineng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," *Proc. SIGSAC*, pp. 1285–1298, 2017.

[4] S. Yang, Y. He, Z. Ge, D. Wang, and J. Xu, "Predictive impact analysis for designing a resilient cellular backhaul network," *SIGMETRICS*, vol. 1, no. 30, 2017.

[5] J. Rafal, Z. Wojciech, and S. Ilya, "An empirical exploration of recurrent network architectures," *Proc. ICML*, pp. 2342–2350, 2015.

[6] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," *Proc. ICLR*, 2017.

[7] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surv. Tutor.*, 2019.

[8] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

[9] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for timeseries mobile sensing data processing," *Proc. WWW*, pp. 351–360, 2017.

[10] T. Kimura, A. Watanabe, T. Toyono, and K. Ishibashi, "Proactive failure detection learning generation patterns of large-scale network logs," *Proc. CNSM 2015*, pp. 8–14, 2015.

[11] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.