Mobile Network Failure Event Detection and Forecasting with Multiple User Activity Data Sets

Motoyuki Oki,[1*] **Koh Takeuchi,**[2*] **Yukio Uematsu**[1]
1. NTT Communications Corporation, Tokyo, Japan
2. NTT Communication Science Laboratories, Kyoto, Japan
{m.ooki, y.uematsu}@ntt.com, takeuchi.koh@lab.ntt.co.jp

Abstract

As the demand for mobile network services increases, immediate detection and forecasting of network failure events have become important problems for service providers. Several event detection approaches have been proposed to tackle these problems by utilizing social data. However, these approaches have not tried to solve event detection and forecasting problems from multiple data sets, such as web access logs and search queries. In this paper, we propose a machine learning approach that incorporates multiple user activity data into detecting and forecasting failure events. Our approach is based on a two-level procedure. First, we introduce a novel feature construction method that treats both the imbalanced label problem and the data sparsity problem of user activity data. Second, we propose a model ensemble method that combines outputs of supervised and unsupervised learning models for each data set and gives accurate predictions of network service outage. We demonstrate the effectiveness of the proposed models by extensive experiments with real-world failure events occurred at a network service provider in Japan and three user activity data sets.

Introduction

Providing stable and high-quality service is a typical mission of mobile network service providers. The amount of data traffic from mobile network services is expected to grow to 49 exabytes per month by 2021, which is seven times the volume of 2016 (Cisco Systems Inc. 2017). However, due to an unexpectedly huge amount of data traffic exceeding network capacity of a provider, a mobile network service experiences severe failures such as network troubles, performance deterioration, and very slow data traffic on rare occasions.

To recover from failures as soon as possible, immediate detection and forecasting of a service outage are crucial problems. Traditional methods for such problems are monitoring network traffic and server logs (Gill, Jain, and Nagappan 2011; Brutlag 2000). However, such methods are not appropriate for our purpose because a modern service with thousands of servers and network appliances around the world and monitoring all of the system equipment is almost impossible (Takeshita, Yokota, and Nishimatsu 2015).

*These two authors contributed equally to this study.

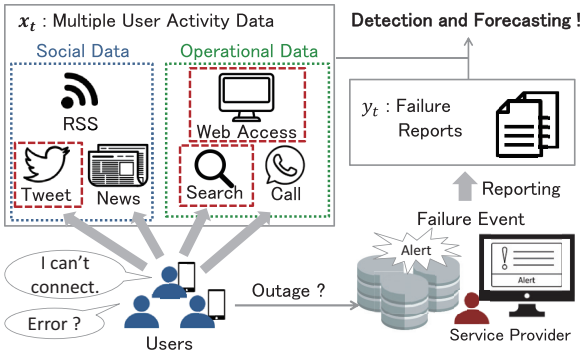

Figure 1: Overall view of a failure event detection and forecasting system with multiple user activity data sets. The data surrounded by the red dotted lines are used in this study.

Mobile network users frequently communicate with a mobile network service and often detect service outage before the service provider detects it (Qiu et al. 2010). They can immediately publish their impressions on the services through social media and search for failure information on the web. There are a few pioneering studies that utilize social data for detecting network failure events (Takeshita, Yokota, and Nishimatsu 2015; Maru et al. 2016). Although they have shown better performances by using supervised learning methods, they have only employed Twitter data and did not consider utilizing other kinds of user activity data such as search queries and web access logs. We illustrate a framework for failure event detection and forecasting for a network service in Fig. 1.

In a field of analyzing social events in the physical world and the web, the event detection or forecast from multiple user activity data sets have been a hot topic (Atefeh and Khreich 2015) and have been applied to various kind of events including planned server downs (Becker et al. 2012), natural hazards (Abhik and Toshniwal 2013), influenza epidemics (Santillana et al. 2015), airport threats (Khandpur et al. 2017), and civil unrest (Kallus 2014; Zhao et al. 2015; 2016). They have demonstrated considerable improvements by simultaneously analyzing multiple user activity data sets because each data represents different aspects of the user behavior. However, to the best of our knowledge, no such study

has focused on network failure events.

Multiple user activity data sets would also be beneficial for detecting and forecasting failure events. For example, we show the number of tweets, web accesses, and search queries including the name of the service for each hour during the four days before and after the occurrence of a failure event in Fig. 2. We also showed typical words, URLs, and queries that frequently appeared in the event. The red region represents the occurrence of the failure event. We can observe unusual but different user activity patterns for each data compared with the same time on different days.

In this paper, we attempt to introduce a network failure event detection and forecasting method based on supervised learning methods that provide a risk of a failure every minute. There are several crucial problems for constructing such a system. **1. Utilizing both social and operational data.** Existing studies on the failure event detection only employed social data and not used operational data. To incorporate these data into our system, we need to develop a method that can successfully take advantage of both social and operational data. **2. Imbalanced label and data sparsity problem.** Failure events rarely occur in a mobile network service. Therefore, almost all of the observed data is labeled as normal, and only a few parts of it are labeled as failure. In addition, observed data aggregated in a very short span such as each minute tend to be sparse. The imbalanced label and sparsity problems often cause the overfitting of supervised learning method and lead poor performances.

To solve these problems, we propose a model ensemble approach with multiple user activity data sets to real-time detections and forecasting failure events in a very short span. Our approach is based on a two-level procedure. First, we propose a novel feature construction method that treats both the imbalanced label problem and the data sparsity problem of the user activity data. Second, to utilize multiple user activity data sets for detecting and forecasting events, we introduce a model ensemble approach that can combine the outputs of supervised and unsupervised learning methods for every data sets to provide accurate predictions. We then propose a novel failure event forecasting method by slightly changing a setting of inputs and outputs of the failure event detection system. We experimentally show the improvement of detection and prediction performance by our proposed approach through extensive experiments using real-world data sets. We demonstrate that the proposed approach by using multiple data outperforms existing methods.

Related work

A supervised failure detection system for an automatic network controlling system has been developed by Maru et al. (2016).One of the most related methods was proposed by Takeshita et al. (2015).Their method combined a keyword matching filter and the support vector machine to extract failure related posts from tweets and gave a failure alert when the number of posts is over a threshold. Those methods have been shown better performances on their problems, but only employed single social data.

In the field of social event analysis, Zhao et al. (2016) proposed a novel multi-source feature learning method based on the logistic regression. Their method consisted of three-level model ensemble approach and showed better performance on the spatio-temporal event forecasting. However, they have only employed linear models for their frameworks. The failure event labels used in these works were provided by annotators who watches a social stream or news, and it is reasonable that they can predict the labels from social data. In contrast, we used a set of labels provided by a service provider and annotated independently from social data. Thus, our problem formulation is different from these of social event analysis.

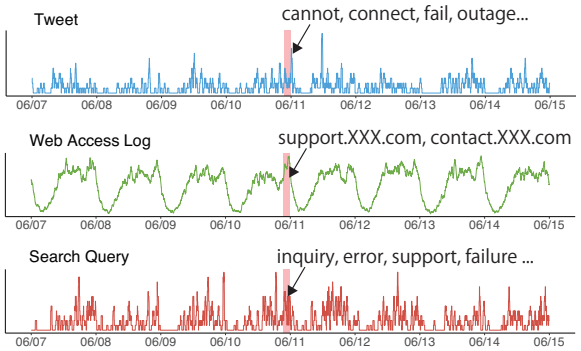

Figure 2: Number of tweets, web accesses of the official pages, and search queries related to the mobile network service in the period before and after a failure event. The red region corresponds to a failure event.

Failure events and user activity data

A service provider announces a failure as a report to inform users when a serious failure event occurred on their mobile network service. We obtained nine serious failure events reported by a mobile network service provider which has millions of subscribers in Japan from November 2015 through December 2016. Those failures were mainly caused by errors in a network system or its applications. The failure event ID, the event duration (time in minutes) and date and time of each ID are shown in Table 1. These events are rare and only occurred once or twice a month. The duration of failure events varied from 60 minutes to 1,554 minutes because every failure event happened on different servers or parts of the system.

We collected one social data and two operational data as user activity data sets from November 2015 to January 2017. As social data, we obtained a set of tweets containing keywords such as the name of the service via Twitter API. The total number of posts was 72,070 which was extremely small compared with common topics tweets such as sports. The nouns, verbs, and adjectives in posts are extracted as words appearing and constructed a set of unique words $D^e = \{d^e_1, \ldots, d^e_{M_e}\}$, where $|M_e| = 5,750$. Stop words, and words whose total frequency was less than three, were removed.Then, we obtained the term frequency per minute for the whole periodto create feature vectors. For convenience, we denote tweet data as $\mathcal{T}$. As operational data, we employ web access logs and search queries on a

Table 1: Failure events published by the network service provider. The duration indicates amount of minutes to recover from failure events. The date and time indicate start day and time of each failure event.

ID	Duration [min]	Date and Time
1	188	Nov/13/2015, 08:22 a.m.
2	60	Feb/04/2016, 06:00 a.m.
3	60	Feb/13/2016, 10:52 p.m.
4	60	Mar/24/2016, 02:00 a.m.
5	150	Jun/10/2016, 09:20 p.m.
6	512	Jun/15/2016, 09:20 p.m.
7	100	Jul/06/2016, 08:50 a.m.
8	950	Jul/27/2016, 10:00 p.m.
9	1554	Dec/25/2016, 01:46 a.m.

Table 2: Training and test data sets. The ratio of failure labels ($y_t = 1$) in the test data set and the sparsity corresponds the proportion of zero elements in the feature vector for the training data sets.

Test ID	Training ID	Ratio [%]	Sparsity [%]		
			$\mathcal{T}$	$\mathcal{W}$	$\mathcal{Q}$
5	1 - 4	0.12	99.55	99.06	99.85
6	2 - 5	0.10	99.56	99.06	99.85
7	2 - 6	0.27	99.57	99.42	99.85
8	2 - 7	0.31	99.58	99.42	99.85
9	4 - 8	0.57	99.58	99.58	99.84

search engine provided by a network service provider in Japan. We obtained 29,520,772 access logs on web sites such as service portal pages, purchase pages, and support pages from the network service provider. An example of a web access log is given as the tuple: (Access Time = "2016-03-20 11:54:12", URL = "http://contact.XXX.com", TITLE = "Contact Form"). We created a set of unique pages $D^w = \{d_1^w, \ldots, d_{M_w}^w\}$, where $|M_w| = 6,378$. Then, we calculated the frequency of access to the pages per minute to create feature vectors. We used 82,136 search queries from the service provider which is provided by a search engine on the Internet. An example of a search query is given by the tuple: (Search Time = "2016-03-21 21:31:56", Query = "XXX error"). We set keywords to be the same as the Twitter data and extracted search queries containing the keywords from all search queries. We created a unique query set $D^q = \{d_1^q, \ldots, d_{M_q}^q\}$ from the space-separated queries, where $|M_q| = 1,209$. We calculated the term frequency in queries per minute in the same way as tweets to create feature vectors. For convenience, we denote web access log data and search query data as $\mathcal{W}$ and $\mathcal{Q}$.

Failure event detection

Problem formulation

$\boldsymbol{x}_t \in \mathbb{R}^M$ is a feature vector of user activity data on the time stamp t and $y_t \in \{-1, 1\}$ is a label that indicates whether an event occurs ($y_t = 1$) or not ($y_t = -1$) at that time. We denote a training data set and a test data set as $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^{T}$ and $\{\boldsymbol{x}_t\}_{t=T+1}^{T+T'}$, respectively, where T and T' are the duration of each data set. Our goal is to estimate a learning model f that predict a label y_t from a feature vector $\boldsymbol{x}_t$.

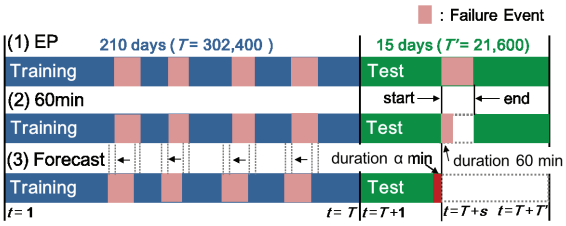

Figure 3: Evaluation settings of AUC for (1) Entire Period (EP), which verify the performance during all periods of test data set, (2) Early detection in 60 minutes (60min), which verify the performance during the 60 minutes after a failure event occur and the rest interval, and (3) Forecasting future events (Forecast), which verify the performance during the periods before a failure event.

We utilized the last five failure events in Table 1 as the test event. For each test event, we constructed a test data set whose start and end time stamps were set at one week before and after the time stamp when the failure event occurred. Since two events (ID:5 and ID:6) occurred within one week, we set test data sets for these events at four days before and after. We employed the previous 210 days from the starting time stamp of the test data set as a training data set. The first four events are only utilized as training data set. We illustrate how we constructed three types of training and test data sets in Fig. 3.

We show the training ID for each test event, the ratio of failure labels ($y_t = 1$) in the test data set, and the sparsity that corresponds the proportion of zero elements in the feature vector for the training data sets in Table 2. As shown in Table 2, our data set contains only less than one percent of labels for failure events. Moreover, the sparsity of every training data is extremely high.

Feature construction for imbalanced labels and sparse time series

The labels in our data set are highly imbalanced, as failure events in the network service rarely occur and are fixed within an hour to a few days. To address the label imbalanced problem, we combine term frequency features with a scaling method that considers the proportion of labels. We adopt a feature scaling called the Bi-Normal Separation (BNS) that was proposed for text classification problems with imbalanced labels (Forman 2003). BNS can be defined as: $\mathrm{bns}(d_i) = |F^{-1}(tpr) - F^{-1}(fpr)|$ where $tpr = tp/(tp + fn)$ and $fpr = fp/(fp + tn)$ indicate true and false positive rate, tp and fn indicate the positive cases containing a feature d_i or not, fp and tn indicate the negative cases, and $F^{-1}(\cdot)$ is the inverse normal cumulative distribution function. To avoid the undefined value $F^{-1}(0)$ and $F^{-1}(1)$, zero and one are substituted by 0.0005 and 0.9995. Then, we obtain feature vectors by multiplying the term frequency vector $tf(x_{i,t}) = \frac{x_{i,t}}{\sum_{m=1}^{M} x_{m,t}}$ with BNS features :

Table 3: Comparison of feature construction methods using individual user activity data by AUC for EP. "tf-bns+sma" outperformed the other methods for three user activity data sets.

Method	$\mathcal{T}$	$\mathcal{W}$	$\mathcal{Q}$
td	$0.52_{\pm 0.03}$	$0.68_{\pm 0.11}$	$0.50_{\pm 0.01}$
tf-idf	$0.53_{\pm 0.03}$	$0.69_{\pm 0.10}$	$0.50_{\pm 0.01}$
tf-bns	$0.53_{\pm 0.03}$	$0.67_{\pm 0.07}$	$0.50_{\pm 0.01}$
td+sma	$0.65_{\pm 0.15}$	$0.82_{\pm 0.16}$	$0.55_{\pm 0.10}$
tf-idf+sma	$0.74_{\pm 0.13}$	$0.88_{\pm 0.10}$	$\mathbf{0.57}_{\pm 0.09}$
tf-bns+sma	$\mathbf{0.77}_{\pm 0.12}$	$\mathbf{0.90}_{\pm 0.10}$	$\mathbf{0.57}_{\pm 0.13}$

$tf\text{-}bns(x_{i,t}) = tf(x_{i,t}) \times bns(d_i)$.

Since our feature vector is highly sparse and often noisy, we employ moving average techniques with a sliding time window to address the sparsity problem (Botezatu et al. 2016). We use a simple moving average for a feature vector $\boldsymbol{x}_t : \overline{\boldsymbol{x}}_t = \sum_{s=s'}^{t} \frac{1}{S} \boldsymbol{x_s}$, where $s' = 1$ if $t - S \leq 0$ and $s' = t - S$ otherwise. We use S as a duration of the sliding time window. We expect that the observations from the past time $t - S$ are used for learning detection models in a time t. A large S is used for assigning weights to observation from the more distant past.

Model comparison

Each user activity data represents a different aspect of user behavior on social media or the operation system. The most suitable model should differ for every data set. Since the relation between the user activity data sets and event labels is not always linear, we utilize several linear and nonlinear supervised classification models including Logistic Regression (LR) (Ramakrishnan et al. 2014), AdaBoost (ADA) of decision stamps (Santillana et al. 2015), Random Forest (RF) (Kallus 2014), and Neural Network (NN). In addition to supervised models, regarding failure events as anomaly events, we also employ unsupervised anomaly detection models One Class SVM (OCS) and Auto Encoder (AE) as comparisons (Chandola, Banerjee, and Kumar 2009).

Experiments and discussion

We examine the Area Under the ROC Curve (AUC) to evaluate the predictive performance of the feature construction methods. We employed "td", which learn the model using the original features $\boldsymbol{x}_t$, "tf-idf" (Salton and McGill 1986), which can be defined as $tf\text{-}idf(x_{i,t}) = tf(x_{i,t}) \times idf(d_i)$ where $idf(d_i) = \log \frac{T}{tp+fp}$, "tf-bns", "td + sma", which applies the moving average to "td", "tf-idf + sma", and "tf-bns + sma". In this experiment, we used a logistic regression model with ridge regularization implementation (sklearn.linear_model.LogisticRegression) provided by scikit-learn (Pedregosa et al. 2011), which is one of the simplest linear models. We selected the regularization parameters $C \in \{0.01, 0.1, 1.0, 10, 100\}$ and the window size $S \in \{10, 30, 50\}$ for the moving average by five-fold cross validation.

The average of AUCs for the Entire Period (EP) of all event IDs (from 5 through 9) in the tweet, web access logs, and search queries are shown in Table 3. Boldface indicates the highest AUC for each user activity data. As shown in Table 3, "tf-bns+sma" provided the best average AUC among all methods. The EP of this method was improved compared to that of "td" by 31 %. We confirmed that improvement by the simple moving average was 19 %. This result supports the effectiveness of the feature scaling by BNS, and the suppression of the sparseness by the simple moving average.

Next, to find the best model for each user activity data, we compared the performance of models on failure event detection. In addition to AUC for EP, we utilize a weighted AUC to evaluate the performance for early detection (60min) because early detection is important for quickly fixing systems (Keilwagen, Grosse, and Grau 2014). For the weighted AUC, we denote a weight on a time stamp w_t and set it to 1 if t is included a non-failure event and 60 minutes after the failure event, and set it to 0 for the others. Feature vectors with "tf-bns + sma" were utilized as input for every data set. The regularization parameter for LR, the number $N \in \{100, 300, 500\}$ of weak classifiers as with decision stamps to obtain RF and ADA and regularization parameter $nu \in \{0.1, 0.5, 0.9\}$ for OCS with Gaussian kernel were determined by five-fold cross validation. We employed an implementation provided by scikit-learn (Pedregosa et al. 2011). For NN and AE, we used the three-layer neural network with M dimension input, a hidden layer with one hundred neurons, ReLU activation function, and Adadelta optimizer. We implemented the two models with Keras. The anomaly score of AE was calculated by the squared reconstruction error.

The results of AUC (EP) and the weighted AUC (60min) are shown in Table 4. "All" indicates the average and standard deviation of AUC of five events. As shown in Table 4, for 60min, AE based on the input of tweet data provided the best average AUC of six models and three data. For EP, LR based on the input of web access logs provided the best average AUC of six models and three data. This result indicates that the early detection based on tweet data performed better than the other two data and the detection in the entire period based on web access logs performed better than the other two data. On the other hand, compared with the models and data in each event, RF based on the input of web access logs and search queries provided the best performance in ID:5 and ID:6 of test data. To sum up these results, the optimal combinations of models and data are different in each event due to the variety of the failure events. In addition, since the score of Twitter data and web access logs were highest in 60min and EP, respectively, this result would suggest an assumption on user behavior that users who detected a failure tend to publish their posts first and then access web pages to obtain failure information. This assumption would shed light on understanding a sequence of user activities on a failure event.

Model ensemble

We propose a two-level model ensemble method for utilizing the multiple user activity data sets. For each data set, we employ LR, RF, and AE for the model ensemble because these models were the best for at least one event ID in the

Table 4: Comparison of models using individual user activity data. "60min" indicates the AUC calculated using during the 60 minutes after a event occur and the rest interval. "EP" indicates the AUC calculated using all periods in test data.The results demonstrate that the best combinations of models and data are different in each event.

Data	Model	ID:5		ID:6		ID:7		ID:8		ID:9		All	
		60min	EP	60min	EP	60min	EP	60min	EP	60min	EP	60min	EP
$\mathcal{T}$	LR	0.92	0.94	0.82	0.60	0.62	0.74	0.99	0.76	**1.00**	0.82	$0.87_{\pm 0.16}$	$0.77_{\pm 0.12}$
	ADA	0.78	0.63	0.02	0.35	0.42	0.45	0.49	0.58	**1.00**	0.51	$0.54_{\pm 0.37}$	$0.50_{\pm 0.11}$
	RF	0.83	0.79	0.72	0.76	0.95	0.96	0.99	0.83	**1.00**	**0.83**	$0.90_{\pm 0.12}$	$0.83_{\pm 0.08}$
	NN	0.79	0.65	0.18	0.34	0.29	0.27	0.86	0.58	0.99	0.47	$0.62_{\pm 0.36}$	$0.46_{\pm 0.16}$
	OCS	0.88	0.69	0.72	0.56	0.50	0.50	0.50	0.56	**1.00**	0.70	$0.72_{\pm 0.22}$	$0.60_{\pm 0.09}$
	AE	**0.94**	0.82	0.86	0.73	0.93	0.90	0.90	0.70	**1.00**	0.73	$\mathbf{0.93}_{\pm 0.05}$	$0.78_{\pm 0.08}$
$\mathcal{W}$	LR	0.78	0.90	0.76	**0.86**	**0.99**	**0.99**	**1.00**	**0.99**	0.98	0.75	$0.90_{\pm 0.12}$	$\mathbf{0.90}_{\pm 0.10}$
	ADA	0.62	0.80	0.86	0.70	0.82	0.89	0.82	0.93	0.94	0.80	$0.81_{\pm 0.12}$	$0.82_{\pm 0.09}$
	RF	**0.94**	**0.97**	0.82	0.75	0.68	0.73	0.93	0.95	**1.00**	0.71	$0.87_{\pm 0.13}$	$0.82_{\pm 0.13}$
	NN	0.57	0.82	0.51	0.71	0.88	0.89	0.98	0.97	0.99	0.75	$0.79_{\pm 0.23}$	$0.83_{\pm 0.11}$
	OCS	0.60	0.50	0.40	0.68	0.79	0.78	0.04	0.51	0.82	0.52	$0.53_{\pm 0.32}$	$0.60_{\pm 0.13}$
	AE	0.41	0.50	0.38	0.69	0.81	0.81	0.03	0.50	0.82	0.47	$0.49_{\pm 0.33}$	$0.59_{\pm 0.15}$
$\mathcal{Q}$	LR	0.53	0.39	0.86	0.58	0.78	0.74	0.45	0.61	0.60	0.55	$0.65_{\pm 0.17}$	$0.57_{\pm 0.13}$
	ADA	0.26	0.33	0.16	0.56	0.61	0.62	0.45	0.53	0.27	0.50	$0.35_{\pm 0.18}$	$0.51_{\pm 0.11}$
	RF	0.73	0.55	**0.87**	0.70	0.62	0.52	0.36	0.64	0.42	0.48	$0.60_{\pm 0.21}$	$0.58_{\pm 0.09}$
	NN	0.25	0.32	0.20	0.51	0.30	0.47	0.54	0.52	0.50	0.52	$0.36_{\pm 0.15}$	$0.47_{\pm 0.09}$
	OCS	0.77	0.68	0.70	0.56	0.66	0.60	0.47	0.57	0.88	0.58	$0.70_{\pm 0.15}$	$0.60_{\pm 0.05}$
	AE	0.78	0.67	0.85	0.62	0.50	0.62	0.37	0.66	0.71	0.57	$0.64_{\pm 0.20}$	$0.63_{\pm 0.04}$

previous experiment. We define our ensemble method as:

$$\text{(Level 1)}\ \ \hat{p}_t = \mathrm{sig}(\boldsymbol{w}^T \boldsymbol{z}_t + b) \tag{1}$$

$$\text{(Level 2)}\ \ \boldsymbol{z}_t = (f^{\mathcal{D}_1}_{\mathrm{M}_1}(\boldsymbol{x}_t), f^{\mathcal{D}_2}_{\mathrm{M}_1}(\boldsymbol{x}_t), \ldots, f^{\mathcal{D}_q}_{\mathrm{M}_3}(\boldsymbol{x}_t)) \tag{2}$$

where sig is the sigmoid function $\mathrm{sig}(a) = \frac{1}{1+exp(-a)}$, $\boldsymbol{w} \in \mathbb{R}^{3\times q}$ is the coefficients for models of Level 2, b is a bias term. We used $f^{\mathcal{D}}_{\mathrm{M}}(\boldsymbol{x}_t)$ and $\boldsymbol{z}_t \in \mathbb{R}^{3\times q}$ as output of three model M_1(=LR), M_2(=RF), and M_3(=AE) with a data set $\mathcal{D}_j, j = (1, \ldots, q)$ in time t and the feature vector for Level 2, respectively. From the definition, the prediction score $\hat{p}_t$ on Level 1 is linearly dependent on Level 2 parameters $\boldsymbol{z}_t$. We define a loss function to estimate the coefficient vector $\boldsymbol{w}$ and the bias term b as:

$$L(\boldsymbol{w}, b) = -\sum_{t=1}^{T} \{(1-y_t)\log(1-p_t) + y_t \log p_t\} + \psi(\boldsymbol{w}, b, C) \tag{3}$$

where $\psi(\boldsymbol{w}, b, C)$ is a regularization term.

We also employ a data ensemble method that trains a detection model $f(\cdot)$ from a feature space with the concatenation of multiple data. We examined the whole combination of user activity data sets in this experiment.

We evaluated AUC for the entire period (EP) and the weighted AUC (60min) for the model ensemble method (ME) and the data ensemble method (DE). Feature vectors with "tf-bns + sma" were used in this experiment. We used a ridge regularization: $\psi(\boldsymbol{w}, b, C) = \frac{C}{2}(||\boldsymbol{w}||^2 + b^2)$. In this experiment, we employed a logistic regression implementation (sklearn.linear_model.LogisticRegression) provided by scikit-learn (Pedregosa et al. 2011). Hyper parameters were selected by five-fold cross validation.

Table 5: Comparison of AUC of model ensemble (denoted as ME) and data ensemble (denoted as DE). The result indicates that the ME with multiple user activity data sets contributed to the improvement from the case where we only used single data.

Data	ME		DE	
	60min	EP	60min	EP
$\mathcal{T}$	$0.94_{\pm 0.09}$	$0.87_{\pm 0.08}$	-	-
$\mathcal{W}$	$0.90_{\pm 0.13}$	$0.88_{\pm 0.11}$	-	-
$\mathcal{Q}$	$0.62_{\pm 0.16}$	$0.60_{\pm 0.08}$	-	-
$(\mathcal{T}, \mathcal{W})$	$\mathbf{0.95}_{\pm 0.10}$	$0.90_{\pm 0.09}$	$0.88_{\pm 0.15}$	$0.88_{\pm 0.09}$
$(\mathcal{T}, \mathcal{Q})$	$0.94_{\pm 0.08}$	$0.87_{\pm 0.08}$	$0.88_{\pm 0.13}$	$0.73_{\pm 0.13}$
$(\mathcal{W}, \mathcal{Q})$	$0.90_{\pm 0.13}$	$0.88_{\pm 0.11}$	$0.90_{\pm 0.13}$	$0.89_{\pm 0.10}$
$(\mathcal{T}, \mathcal{W}, \mathcal{Q})$	$0.94_{\pm 0.09}$	$\mathbf{0.91}_{\pm 0.09}$	$0.88_{\pm 0.14}$	$0.88_{\pm 0.10}$

The comparison of AUC of model and data ensemble are shown in Table 5. Boldface indicates the highest AUC for each evaluation. We confirmed that the model ensemble with tweet and web access logs ($\mathcal{T}$,$\mathcal{W}$) and the whole data set ($\mathcal{T}$,$\mathcal{W}$,$\mathcal{Q}$) achieved the best score for 60min and EP, respectively. The result indicates that the model ensemble method with multiple user activity data sets contributed to the improvement from the case where we only used single data. On the other hand, as shown in Table 5, the data ensemble method could not improve the performance. The increase of the feature dimension might cause the overfitting for the model.

An example of prediction scores and AUC for the entire period of our model ensemble method and the best method for every individual data set is shown in Fig. 4. We confirmed that the model ensemble successfully suppressed the false negative, e.g. the prediction score in web access logs in Dec/31/2016, in the period before and after the failure

Table 6: Comparison of forecasting performance in α minutes. Model ensemble method outperforms the other methods for the failure event forecasting.

Model	$\alpha = 10$	$\alpha = 20$	$\alpha = 30$	$\alpha = 40$	$\alpha = 50$	$\alpha = 60$
ME($\mathcal{T}, \mathcal{W}$)	$\mathbf{0.81}_{\pm 0.27}$	$\mathbf{0.79}_{\pm 0.25}$	$\mathbf{0.78}_{\pm 0.21}$	$\mathbf{0.74}_{\pm 0.18}$	$0.69_{\pm 0.18}$	$0.67_{\pm 0.19}$
ME($\mathcal{T}$)	$0.72_{\pm 0.35}$	$0.71_{\pm 0.36}$	$0.71_{\pm 0.34}$	$0.68_{\pm 0.31}$	$0.72_{\pm 0.20}$	$\mathbf{0.76}_{\pm 0.15}$
ME($\mathcal{W}$)	$0.76_{\pm 0.25}$	$0.74_{\pm 0.22}$	$0.77_{\pm 0.17}$	$0.70_{\pm 0.16}$	$\mathbf{0.77}_{\pm 0.09}$	$0.70_{\pm 0.24}$
DE($\mathcal{T}, \mathcal{W}$)	$0.72_{\pm 0.35}$	$0.69_{\pm 0.31}$	$0.66_{\pm 0.26}$	$0.64_{\pm 0.18}$	$0.69_{\pm 0.08}$	$0.68_{\pm 0.12}$
LR($\mathcal{T}$)	$0.79_{\pm 0.27}$	$0.67_{\pm 0.31}$	$0.60_{\pm 0.29}$	$0.54_{\pm 0.25}$	$0.48_{\pm 0.23}$	$0.49_{\pm 0.19}$
LR($\mathcal{W}$)	$0.70_{\pm 0.25}$	$0.71_{\pm 0.27}$	$0.71_{\pm 0.25}$	$0.73_{\pm 0.22}$	$0.73_{\pm 0.20}$	$0.73_{\pm 0.18}$

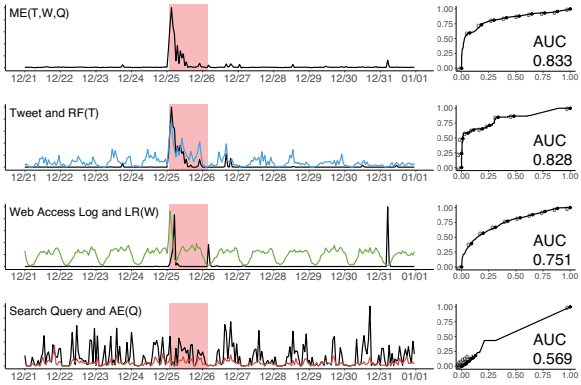

Figure 4: Performance and observed user activities of the test ID:9. The model ensemble ME($\mathcal{T}$,$\mathcal{W}$,$\mathcal{Q}$) outperformed RF($\mathcal{T}$), LR($\mathcal{W}$), and AE($\mathcal{Q}$). Black lines indicate the prediction score. Blue, green, and red colored line indicate the number of tweets, web access logs, and search queries. Figures in the right side correspond the AUC for the entire period of the test data set.

event occurred. For the failure period, our method provided a rapid rise and long-tailed decreasing of the prediction score because it can exploit both the rapid rise of scores on tweet data and the delayed rise of web access logs.

Failure event forecasting

We address the future failure event forecasting problem in this section. We denote a training set of user activity data as $\{(\boldsymbol{x}_t, y_{t+\alpha})\}_{t=1}^{T-\alpha}$, where T is a time stamp 210 days + α minutes before the failure event started. Thus, the problem is to predict the failure event labels on α minutes later. To evaluate the predictive performance, we set the test set as $\{\boldsymbol{x}_t\}_{t=T-\alpha+1}^{T+s}$. We employed the model ensemble method with tweets and web access logs in this experiment. We used LR with tweets or web access logs (LR($\mathcal{T}$) and LR($\mathcal{W}$)), the data ensemble method with tweets and web access logs (DE($\mathcal{T}, \mathcal{W}$)), and the model ensemble methods with Twitter or web access logs (ME($\mathcal{T}$) and ME($\mathcal{W}$)) for comparisons. The time step α were set to $(10, 20, 30, 40, 50, 60)$. To evaluate the predictive performance, we calculated AUC for the forecasting, see Fig. 3.

The average and standard deviations of AUC for each time step α are shown in Table 6. Boldface indicates the best performance for each time step. ME($\mathcal{T}$,$\mathcal{W}$) showed the best performance for $\alpha = 10, 20, 30,$ and 40. ME($\mathcal{T}$) and ME($\mathcal{W}$) showed the best performance for $\alpha = 50$ and $\alpha = 60$. On the other hand, LR($\mathcal{T}$) and LR($\mathcal{W}$) resulted in lower AUC than that of ME. This result indicates that our model ensemble method was more highly effective than existing methods for the failure event forecasting with both multiple and individual user activity data.

Conclusion

We proposed a feature construction method and model ensemble method for detecting and forecasting failure events on mobile network services from multiple user activity data sets. Because we would like to help service providers to take quick measures for fixing their system, we conducted two types of extensive experiments that considered the performance on (1) stable failure event detection on the entire test period and early failure detection in 60 minutes (2) beforehand forecasting of a failure event. With real-world failure events, we demonstrated that our proposed approaches outperformed existing methods utilizing only single user activity data set.

There are promising directions for future works below. We employed three types of user activity data sets in this study. Utilizing additional data sets that we mentioned in Fig. 1 or something else would be beneficial for providing more accurate predictions. According to wide movements on neural networks, a nonlinear method for time series analysis would be useful for our target. Moreover, analyzing failure symptoms in each event and deploying a monitoring system based on our ensemble approaches using multiple user activity data would be important future works.

Acknowledgments

We would like to thank Dr. Naonori Ueda and Akinori Fujino for their supports and helpful discussions on this study.

References

Abhik, D., and Toshniwal, D. 2013. Sub-event detection during natural hazards using features of social media data. In *WWW*.

Atefeh, F., and Khreich, W. 2015. A survey of techniques for event detection in twitter. *Journal Computational Intelligence* 31(1):132–164.

Becker, H.; Iter, D.; Naaman, M.; and Gravano, L. 2012. Identifying content for planned events across social media sites. In *WSDM*.

Botezatu, M.; Giurgiu, I.; Bogojeska, J.; and Wiesmann, D. 2016. Predicting disk replacement towards reliable data centers. In *SIGKDD*.

Brutlag, J. D. 2000. Aberrant behavior detection in time series for network monitoring. In *LISA*, 139–146.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41(3):1–72.

Cisco Systems Inc. 2017. Cisco visual networking index: Global mobile data traffic forecast update, 2016 - 2021.

Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research* 3:1289–1305.

Gill, P.; Jain, N.; and Nagappan, N. 2011. Understanding network failures in data centers: measurement, analysis, and implications. In *SIGCOM*.

Kallus, N. 2014. Predicting crowd behavior with big public data. In *WWW*.

Keilwagen, J.; Grosse, I.; and Grau, J. 2014. Area under precision-recall curves for weighted and unweighted data. *PLoS ONE* 9(3).

Khandpur, R. P.; Ji, T.; Ning, Y.; and et al. 2017. Determining relative airport threats from news and social media. In *IAAI*.

Maru, C.; Enoki, M.; Nakao, A.; Yamamoto, S.; Yamaguchi, S.; and Oguchi, M. 2016. Development of failure detection system for network control using collective intelligence of social networking service in large-scale disasters. In *HT*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Qiu, T.; Feng, J.; Ge, Z.; Wang, J.; Xu, J.; and Yates, J. 2010. Listen to me if you can : Tracking user experience of mobile network social media. In *IMC*.

Ramakrishnan, N.; Butler, P.; Muthiah, S.; and et al. 2014. 'beating the news' with embers: forecasting civil unrest using open source indicators. In *SIGKDD*.

Salton, G., and McGill, M. J. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.

Santillana, M.; Nguyen, A. T.; Dredze, M.; M. J. Paul, E. O. N.; and Brownstein, J. S. 2015. Combining search, social media, and traditional data sources to improve influenza surveillance. *PLoS Comput Biol* 11(10).

Takeshita, K.; Yokota, M.; and Nishimatsu, K. 2015. Early network failure detection system by analyzing twitter data. In *IM*.

Zhao, L.; Sun, Q.; Ye, J.; Chen, F.; Lu, C.-T.; and Ramakrishnan, N. 2015. Multi-task learning for spatio-temporal event forecasting. In *SIGKDD*.

Zhao, L.; Ye, J.; Chen, F.; Lu, C. T.; and Ramakrishnan, N. 2016. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *SIGKDD*.