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, 01:40 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 $\boldsymbol{x}_t$.

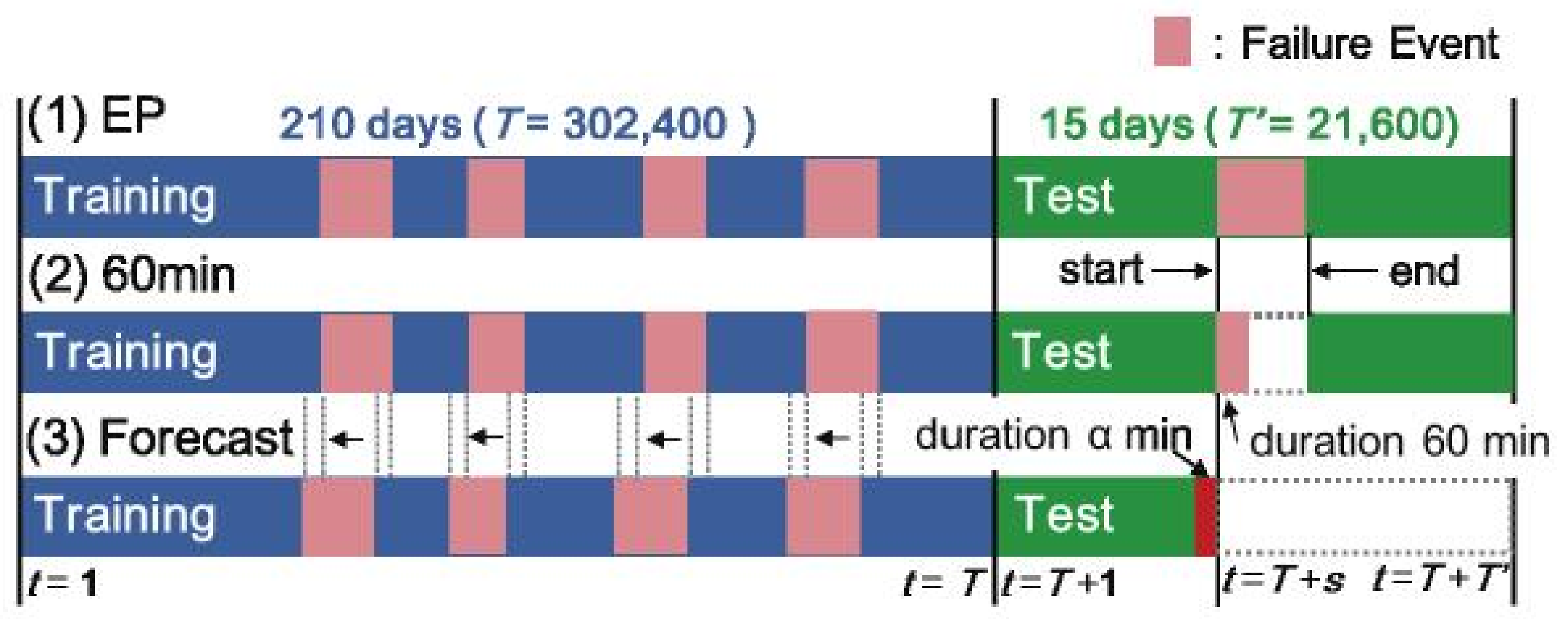

Figure 3: Evaluation settings of AUC for (1) Entire Period (EP), which verify the performance during all periods of test data set, (2) Early detection in 60 minutes (60min), which verify the performance during the 60 minutes after a failure event occur and the rest interval, and (3) Forecasting future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