Journal of Communications and Information Networks, Vol.3, No.3, Sept. 2018
DOI: 10.1007/s41650-018-0025-2

Research paper

Data-Driven User Complaint Prediction for Mobile Access Networks

Huimin Pan, Sheng Zhou, Yunjian Jia, Zhisheng Niu, Meng Zheng, Lu Geng

***Abstract*—In this paper, we present a user-complaint prediction system for mobile access networks based on network monitoring data. By applying machine-learning models, the proposed system can relate user complaints to network performance indicators, alarm reports in a data-driven fashion, and predict the complaint events in a fine-grained spatial area within a specific time window. The proposed system harnesses several special designs to deal with the specialty in complaint prediction; complaint bursts are extracted using linear filtering and threshold detection to reduce the noisy fluctuation in raw complaint events. A fuzzy gridding method is also proposed to resolve the inaccuracy in verbally described complaint locations. Furthermore, we combine up-sampling with down-sampling to combat the severe skewness towards negative samples. The proposed system is evaluated using a real dataset collected from a major Chinese mobile operator, in which, events due to complaint bursts account approximately for only 0.3% of all recorded events. Results show that our system can detect 30% of complaint bursts 3 h ahead with more than 80% precision. This will achieve a corresponding proportion of quality of experience improvement if all predicted complaint events can be handled in advance through proper network maintenance.**

***Keywords*—data-driven complaint prediction, complaint location, network management, machine learning pipeline**

Manuscript received May 31, 2018; accepted Jul. 24, 2018. This work is sponsored in part by the National Natural Science Foundation of China (Nos. 91638204, 61571265, 61621091), and Hitachi Ltd. The associate editor coordinating the review of this paper and approving it for publication was X. Cheng.

Huimin Pan, Sheng Zhou, Zhisheng Niu. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: sheng.zhou@tsinghua.edu.cn)

Yunjian Jia. College of Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: yunjian@cqu.edu.cn)

Meng Zheng, Lu Geng. Hitachi (China) Research & Development Cooperation, Beijing 100190, China

I. Introduction

With the proliferation of smart devices such as smart phones and tablets, it has become essential for people to be able to enjoy mobile services anywhere and anytime. To meet this desire, there has been significant growth in the number of mobile users and traffic over the past few years. At the same time, the service quality required from mobile networks has also changed, and the emerging machine-to-machine traffic have distinct requirements with respect to bandwidth and latency. To address the increasing demands, mobile networks are also evolving at an unprecedented rapid pace. The next-generation mobile access networks will incorporate a wide range of new technologies and architectures, including massive multi-input multi-output (MIMO), millimeter wave, and cloud-based networking.

However, this trend will inevitably lead to increasingly complex networks. The quality of experience (QoE) that is perceived by mobile users will be influenced by numerous network functions, both physical and virtual, which also interact in complex ways. Owing to the complexity, it will become extremely challenging for mobile operators to maintain the health of their networks using only traditional human monitoring and maintenance. In contrast, the autonomous operation of mobile networks will be more responsive, comprehensive, and cost-efficient than human-centric methods. Consequently, the automatic monitoring, reporting, and maintenance of mobile networks have become both important and valuable.

The deployment of active end-to-end probes across the network is a widely used method to prevent the occurrence of network anomalies[1,2]. By injecting probe packets into network and monitoring transmission processes using deployed probes, potential disruptions may be detected. However, with the rapid increase in network complexity[3], the use of probes to cover all key combination nodes is tedious. On the other hand, the importing of additional packets may change the raw performance of the networks, or may even affect mobile users' service experiences, which is contrary to the original purpose.

Passive monitoring-based systems on both the end-user side[4] and service-provider side[5,6] are used to monitor the underlying traffic. Using these systems, any abnormal behav-

ior in traffic patterns is related to network anomalies, and supports performance maintenance without introducing interference information. The major advantage of this kind of method is the assumption that the reduction in the QoE of users will result in a decrease in the traffic load, which is very challenging. Many technical approaches such as network offloading have been used to handle the network traffic demand. Therefore, even if the load decreases in an abnormal manner, the users' service experience may not be impacted. In order to guarantee a good user-service experience, traffic monitoring is not considered an efficient way to deal with network anomaly.

Fueled by recent developments in data platforms and machine-learning algorithms, data-driven methods have become an attractive alternative to autonomous network operation, and they have attracted much attention. Ref. [8] summarizes the literature involving machine-learning algorithms that are applied to self-organizing cellular networks. The k-NN algorithm is used to address user mobility in Ref. [9]. A crowd-sensing method is built to suppress fake sensing attacks in Ref. [10]. In the field of computer networks, statistical modeling and neural-network classification have been widely applied to network intrusion and fault-detection applications[11-13]. However, most of these studies focus on improving the security of computer systems and networks, which means that most of the anomalies are generated by hostile attacks. Therefore, these methods may not be suitable for anomalies that are caused by network resource constraints and instability.

External datasets are considered to be an additional source of information for network anomaly detection. Ref. [14] proposes a usage-based method to detect mobile network failure by monitoring aggregate customer usage. Ref. [15] tries to predict users' complaints in Internet protocol television (IPTV) networks. As in IPTV networks, data are all collected from users' set-top boxes, and it is easy to correlate user complaints with unique set-top boxes, which makes the location of anomalies simple. However, this may not be suitable in mobile access networks.

Although previous studies have demonstrated their effectiveness in the detection of network anomalies, a major drawback is that the detection results cannot be easily related to the network QoE. Self-organizing functions in current systems can automatically mitigate the effect of some network anomalies, leaving negligible degradation in actual user experiences. For example, traffic offloading can automatically dissipate the congesting traffic into neighboring cells before users are made aware of the problem. However, users may not utilize the malfunctioning network element when anomalies occur happen. These hidden anomalies will not hurt the user experience either. Owing to this mismatch, the effort to optimize networks according to detected anomalies may not result in fruitful QoE improvement. In order to improve QoE more effectively, we believe that the detection process should keep the user experience in the loop. To do so, indicators of the user experience should be incorporated into the detection target. In this way, the relationship between anomalies and the user experience can be modeled and then used to weigh the severity of anomalies, and to help mobile operators prioritize counter-measures.

In this paper, we propose the platform for advanced network data analytics (PANDA) following such an argument. PANDA can predict whether or not user complaints will surge in a fine-grained spatial area within a given time window. The mobile operators can then diagnose and fix the base stations (BSs) in the detected area in order to mitigate complaint bursts in advance. We form the prediction model in PANDA following a data-driven fashion: network-monitoring data and user-complaint records are transformed into features and targets of a machine-learning pipeline, and this generalizes the relationship between the complaint and the network monitoring data. For the dataset, we use different forms that require special treatment before they can be used by machine-learning models. We propose a fuzzy spatial gridding method to combat the inaccuracy in complaint location. We also extract complaint burst events with filtering to avoid the noise in raw complaint time series. Moreover, we apply a multi-scale time windowing method to distill temporal features. PANDA is evaluated using a real dataset collected from a major Chinese mobile network. The dataset exhibits a high skewness with only 0.3% of events in complaint bursts. PANDA can recall 30% of these events in complaint bursts 3 h ahead with an 80% accuracy. The contributions of this paper are as follows:

- We propose to incorporate QoE indicators when detecting anomalies in mobile networks. Following this idea, we propose PANDA, which can predict the location and time of future complaint burst events by mining through network monitoring data and complaint records.
- We design several measures to deal with the special problems in complaint prediction. We use a fuzzy spatial gridding method to resolve the inaccuracy in verbally described complaint locations, and we apply filtering on raw complaint events to reduce the noise and obtain complaint burst events; we also propose to use multi-scale time windowing to extract fine temporal features from the regulated data sources.
- We evaluate the proposed system using real-life data collected from a major Chinese mobile network. Our proposed system can recall 30% of these events in complaint bursts 3 h ahead with an 80% accuracy in the highly imbalanced dataset. We also discuss the influence of various system parameters.

The rest of the paper is organized as follows: In section II, we analyze the characteristics of complaint events. Details of our prediction system are described in section III. The proposed system is evaluated based on underlying network data in section IV. Finally, the paper is concluded in section V.

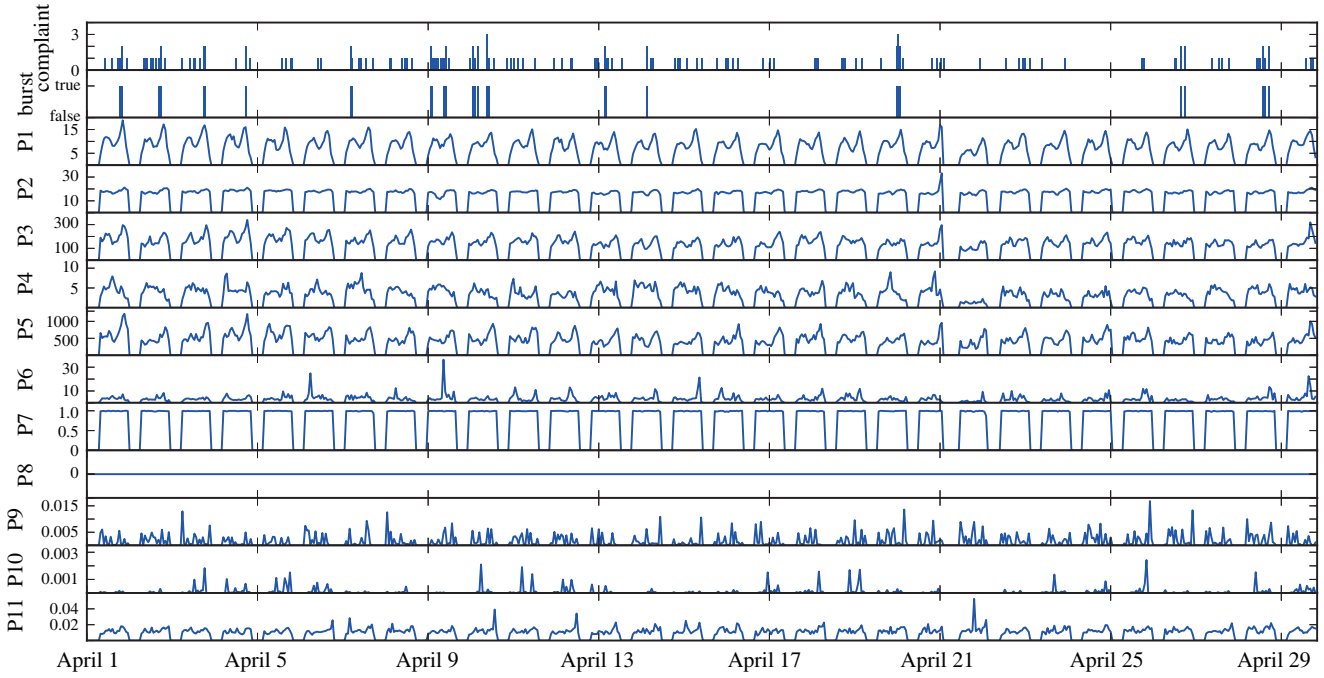


Figure 1 Hourly-average time series of user complaints (top), complaint bursts (second), and 10 BS performance indicators (rest) in a typical 3-base-station spatial grid during April, 2015

II. Understanding User Complaint Prediction

Despite its importance, complaint predictions are significantly complicated by three distinguished characteristics of user complaints. In this section, we describe these characteristics with the help of data visualization. Then, we explain the challenges that they impose, and propose methods to tackle the challenges.

A. Uncertain Root Causes

In general, users' complaints reflect their dissatisfaction with the network's QoS. However, the uncertainty in human behavior can significantly attenuate the correlation between a single piece of complaint and the system anomaly that triggered it in the first place. For example, because contacting customer service staff can be time-consuming, users may be reluctant to call in when the network problem is only slightly annoying. Consequently, the corresponding complaints are likely to be delayed to an unknown time, or they may even be completely abandoned. Such micro-scale uncertainty of individual users is aggregated at the macroscale and transformed into noise-like series of complaints, as shown in the top-most sub-figure of Fig. 1 (The performance indicators in the figure are described in Tab. 1). Hence, the prediction of a single complaint can be extremely difficult and unrewarding.

In contrast, users' responses tend to be quick and intense under serious system malfunctions: if the cellular service in an area is severely disrupted owing to a power black-out, a large number of user complaints can be expected afterwards. Owing to the negative impact of such incidents, some network operators consider the frequency of complaint surges as one of their key performance indicators (KPIs). With regard to the aforementioned issue, we also focus our system on predicting such surges in user complaints. We filter the time series of user complaints, and extract complaint burst events during which time the number of user complaints is suspiciously high. These burst events, as shown in the second-to-top sub-figure of Fig. 1, are used as the prediction targets. Note that there is a chance that fake surges that are purely due to random coincidence are also labeled positive. Such labeling noise can

Table 1 Description of the 10 BS performance indicators

index	category	description
1	TCH	voice channel aggregate traffic
2	EDGE	equivalent data traffic
3	EDGE	upstream EGPRS traffic
4	EDGE	upstream GPRS traffic
5	EDGE	downstream EGPRS traffic
6	EDGE	downstream GPRS traffic
7	TCH	wireless access
8	TCH	voice channel dropping rate
9	SDCCH	SCH congestion rate
10	SDCCH	SCH dropping rate

be reduced by more conservative labeling or manual expert cleansing.

B. Inaccurate Localization

Owing to the cellular nature of mobile communication systems, the localization of user complaints is necessary to identify the responsible network element. Nonetheless, location information provided in real life is often too inaccurate to be used directly: complaint location is most commonly described verbally by users during complaint calls, and is logged into the system as formatted address strings. These addresses are often so coarse that BSs are within the vicinity. Further, even if operators are equipped with instrumentation to capture the cell from which the users issued complaint calls, users can still be some distance away from the responsible cell, whether owing to user mobility or to find a better signal.

We employ a fuzzy association method to cope with the inaccuracy in location information. First, we obtain the geographical coordinates of a part of a complaint by looking up its address in a geographic information system (GIS) database. Then, we blame it on a spatial grid determined by its k-nearest BSs, the exact coordinates of which are known beforehand. A prerequisite of this method is that the inaccuracy of localization is constrained to a small geographical span. If recorded addresses are too coarse or even completely wrong, our proposed method may associate a complaint wrongly with spatial grids, creating false input.

C. Weak Correlation

As described above, we establish a predictive relationship between network monitoring data and complaint burst events. To determine the feasibility of the naive linear predictor, we calculate the linear correlation coefficients between the time series of complaints and 10 BS performance indicators, respectively. The histogram of coefficients across spatial grids is shown in Fig. 2 (Only the grids with more than 10 complaints during the studied period are considered). As can be seen in the figure, the correlation is quite low, indicating that a naive linear predictor may be incapable of the intended task. This motivates us to turn to machine-learning models to fully exploit the potential non-linear correlation, and to build an effective predictor.

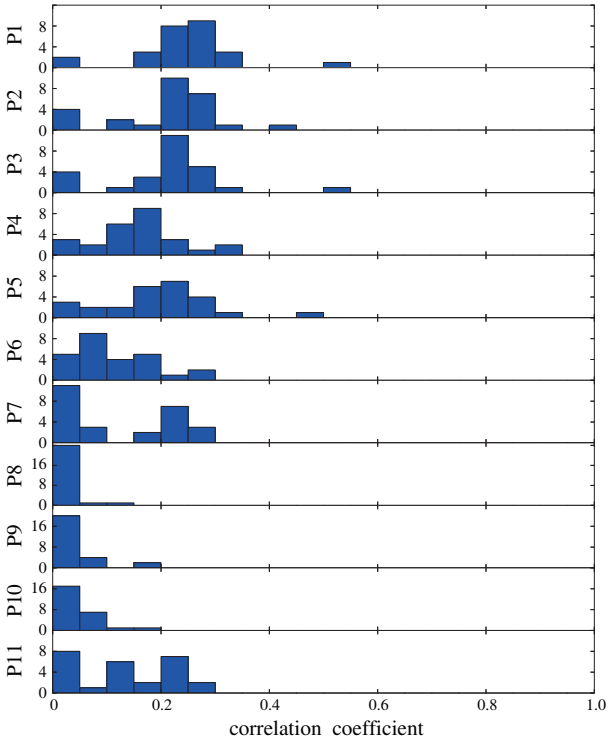


Figure 2 Histogram of linear correlation coefficients between complaint and 10 BS performance indicator time series across different spatial grids

III. System Design

In this section, we introduce the design of the proposed system. First, we give an overview of the system structure, and then we describe the details of each pipeline step.

A. Overview

The structure of the proposed system is illustrated in Fig. 3. The inputs (bottom-left) to the system are data logs that contain hourly network indicator reports, system alarm logs, and user-complaint records; while the output (right-most) is a prediction of whether a complaint burst event is about to happen in a spatial grid during a future time window. The operation of the system is divided into an offline training phase, in which a prediction model is formulated based on historical data, and an online prediction phase which performs the actual prediction based on real-time input. Both phases are composed of multiple data-preparation steps and a machine-learning pipeline.

B. Preparation Steps

The data-preparation steps first combine complaint data together with network monitoring data based on the proposed fuzzy gridding method. They then transform the gridded complaint events into complaint burst events, and extract feature vectors from the merged data through multiscale windowing. Missing data points are also filtered out during the process. The prepared data are finally presented in fine matrix form to the machine-learning pipeline for either offline training or online prediction. Next, we describe each preparation step in detail.

1) *Spatial Gridding and Time Binning:* As described above, one major difficulty in processing user complaint records lies in their inaccurate localization. Therefore, although we know the exact geographical location of BSs, this piece of information cannot be used in a straightforward way

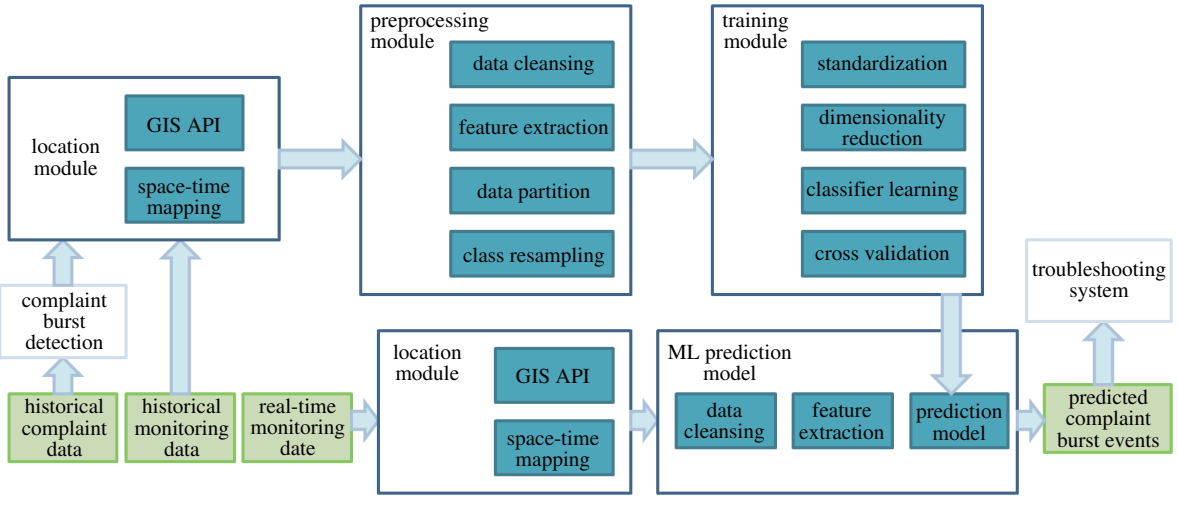


Figure 3 Diagram of the proposed complaint prediction system

to associate them with complaint records.

As a solution, we assume that the inaccuracy introduced by verbal description is locally constrained, so we can resolve the uncertainty by fuzzily associating a complaint event with multiple network elements. More specifically, for each section of a user complaint, we extract the recorded address string and find the corresponding geographical coordinates using GIS API. The coordinates are then compared with that of BSs to find the k nearest one; ties are broken deterministically. In this way, the k nearest BSs can unambiguously determine the geographical area to which the complaint is most likely belong. We denote each of these areas with a unique spatial ID (SID) that is derived from the IDs of its composing BSs. After fuzzy gridding, the performance indicator and alarm logs of BSs can then be associated with spatial grids in a straightforward manner.

It is difficult to analytically derive all of the possible spatial grids because their existence depends largely on the geographical layout of BSs as well as the k used. For this reason, we choose to generate spatial grids as we process complaint records, generating a new grid only when the k nearest BSs form a new one. Note that this method will overlook spatial grids that do not possess any of the recorded complaints. However, given that the records cover a sufficiently long period, it is also reasonable to argue that these grids are unlikely to cause serious complaint surges in the future. Therefore, we can reduce the complexity of gridding by excluding many true negatives at the cost of a few false negatives.

The data input for PANDA may also have inconsistent time formats, e.g., performance indicators are provided hourly, while complaint records and alarm logs come with a second-level time-stamp. We applied time binning to unify the time formats. Specifically, a binning granularity that is no finer than the most coarse input is first chosen (in our case, no finer than 1 h), and then data entries are assigned accordingly to discrete bins with unique time IDs (TIDs). Further, data entries with the same TID and SID are aggregated using some predefined function, e.g., summing or averaging.

2) *Complaint Burst Detection:* As explained in the previous section, we chose the occurrence of complaint burst events as our prediction target. For the implementation, we applied a straightforward method of low-pass filtering followed by thresholding. Given the discrete complaint time series of a spatial grid, we first passed the time series through a linear filter with a short impulse response, e.g., $\{\frac{1}{2}, 1, \frac{1}{2}\}$. Then, we compared the filtered time series against a threshold value, and the time bins in which the filtered value exceeds the threshold are considered to have a complaint burst. We chose this method because it can identify two important temporal complaint patterns, namely a very intense complaint in a single time bin, or many intermittent complaints in consecutive time bins, which are the most dominant complaint patterns from our observation. There are other ways of defining a complaint burst event, e.g., based on the empirical complaint intensity distribution. We chose the proposed method because it can readily give very intuitive results with much less computational complexity.

3) *Feature and Target Generation:* After the above procedures, the multiple raw data sources (performance indicators, alarms logs, complaint events, and burst events) are reduced to fixed-length vectors with a common set of (SID, TID) keys. To generate feature vectors and prediction targets for the subsequent machine-learning pipeline, we further employed multiscale windowing on the data vectors. We first specify N_f feature time windows $W_f = \{(s_f^i, e_f^i), i = 1, 2, \cdots, N_f\}$ and one target time window $w_t = (s_t, e_t)$, with each tuple representing each window's relative offset (edge included) to a refer-

ence TID. For convenience, we assume the values of feature time window to be the negative of their actual offset values, and $s_f^i \leqslant e_f^i$, while target time windows use the original offset value. Note that to guarantee causality, feature time windows should be strictly earlier than the target time window. A safeguard for this requirement is to use positive s_f^i and a zero s_t because the actual value of the reference TID does not really matter.

For each SID and each feature time window, we determined the element-wise average for all of the raw data vectors whose TID falls within that time window, resulting in a new vector having the same dimensionality as the original raw data vector(s). The average vectors corresponding to all combinations of feature time windows and input data sources are then concatenated to form the final feature vector sample. In the training phase, we also need to create a prediction target for each feature vector. Here, we label a feature vector as positive if the TID of any complaint burst event falls within the target time window, and consider it negative for other cases. Finally, the feature vectors as well as labels (only for the training phase) are passed to the machine-learning pipeline.

4) *Data Cleansing:* Occasionally, the raw data vector may be missing for some (SID, TID), whether owing to malfunctions in the logging system or data loss. To guarantee that missing data do not introduce noise into the feature vector, we applied data cleansing at the same time with feature generation. The basic logic is that the number of raw vectors to be averaged should be equal to the length of the time window without missing data. Thus, if we observe otherwise, we can decide that the time window is compromised and should be filtered out. In the end, we only preserve feature vectors for which none of the feature time windows encountered missing data.

C. Machine-Learning Pipeline

We feed the labeled feature vectors into a machine-learning pipeline for training and prediction. The pipeline contains steps for standardization, class resampling, dimensionality reduction as well as a classifier. In the training phase, the pipeline is also cross-validated with a time-based splitting method to prevent overfitting.

1) *Standardization:* Because the feature vectors are derived from a set of heterogeneous raw data vectors, each of the feature dimensions may have a significantly different bias and scale of variance. Dimensions with an overwhelmingly high variance may saturate the classifier. To avoid this, we standardize each dense feature dimension to zero mean and unit variance using linear transformation and scaling. For sparse dimensions, we only scale them to unit variance to avoid breaking their sparsity and dramatically increasing the PANDA's memory footprint.

2) *Class Resampling:* Complaint prediction is a typical imbalanced prediction task: there are far fewer positive samples than negative samples. A standard approach to combat imbalanced datasets is to use class resampling. We can either randomly repeat the minority positive samples (up-sampling) or randomly discard some of the majority negative samples (down-sampling). Both methods are known to have drawbacks: up-sampling will introduce artificial patterns by distorting the positive distribution, while down-sampling will reduce the amount of available data. Hence, we choose to combine these two methods by applying up-sampling and down-sampling with the prescribed order and relative ratio in order to achieve the target ratio of positive and negative samples.

3) *Dimensionality Reduction:* The number of feature dimensions is proportional to the number of feature time windows and raw data dimensions. Therefore, it is possible that we introduced too many feature dimensions and stumbled the classifier (curse of dimensionality). To solve this problem, we also included an (optional) principal component analysis (PCA-) based dimensionality reduction step to help reduce the dimensionality in unfavorable cases.

4) *Machine-Learning Classifier:* We mainly rely on a machine-learning classifier to derive the non-linear relationship between the target and the feature vector. We experimented with logistic regression (LR), decision trees (DTs), random forests (RFs), and support vector machine (SVMs). We find RF to be best suited to our problem because of the large data size and high noise level. Details are presented later in the Evaluation section.

5) *Cross-Validation:* In cross-validation, it is important to avoid data leakage from the validation set into the training set. This may occur if the target time window of validation samples overlaps with the feature time window of training samples in our problem. To avoid this problem, we applied a time-based splitting method. The labeled training vectors are classified into training and validation sets based on their reference TID: samples before the splitting TID are classified into the training set, while the others are put into the validation set. If an additional test set is needed, we randomly draw samples from the validation set to form one.

IV. Evaluation

In this section, we evaluate the proposed PANDA system based on a real dataset. We start by describing the dataset used, then introduce our basic parameter setting and evaluation standard. Finally, we present the performance of PANDA under different meta parameters. Complaint indicators are described in Tab. 2.

Table 2 Description on complaint indicators

indicator	description
k	gridding granularity
s_f	start time of a feature time window
e_f	end time of a feature time window
s_t	start time of a target time window
e_t	end time of a target time window
tw	time window
r_{PN}	ratio of positive samples versus negative samples

A. Dataset Description

The evaluation dataset is collected from a second-tier city in western China by a major Chinese mobile operator. On the user side, it contains the complaint records from all mobile users in the city, while on the network-side, it contains performance indicator reports and alarm logs of 2G BSs city-wide as well as static information about their configuration; all of the data except for the performance indicator reports cover the period from January 2015 to May 2015, while the performance indicators are only provided for April 2015. The raw dataset amounts to around 6GB in total.

The complaint dataset contains multiple information fields, including the time-stamp of the complaint call, the logical category of the reported problem, the address of the complainant, and other miscellaneous fields. The complaint category is chosen by the customer service representative from a pre-compiled list of categories. In the evaluation, we only extracted the complaint categories that pertain to the more serious problems of coverage quality, voice calls, and mobile data services. The BS performance indicator dataset exhibits hourly reports of 10 core indicators, such as aggregated uplink and downlink traffic, congestion rate, and call drop rate. The equipment alarm log records the responsible network element, i.e., time-stamp, category, and the severity of system alarms. There is a total of more than 200 alarm categories, each of which is used as a dimension when generating the raw data vectors. Owing to the high dimensionality, the alarm vector is stored as a sparse vector during implementation.

B. System Configuration

As discussed in previous sections, we can achieve different tradeoffs between the inaccuracy of the complaint location and the uncertainty of the responsible BS(s) by varying the gridding granularity, k. If k is set to be too small, for instance 1, complaint events are in danger of being wrongly blamed by a BS neighboring the responsible one; however, if k is set to be too large, the prediction will become largely unusable because we have to go through each of the k BSs to find the responsible one. Therefore, we set k to an intermediate value of 3 in the evaluation.

C. Evaluation Results

PANDA has numerous tunable parameters besides the basic ones described above. In this section, we experiment with the parameters that have the most significant influence on the system performance, and we discuss their implications. These parameters include feature time window, target time window, re-sampling ratio, and the choice of machine-learning classifier.

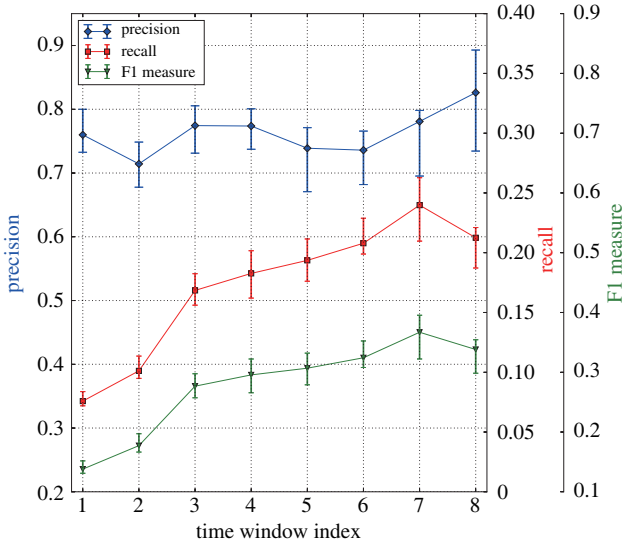
1) *Evaluation Standard:* There are numerous options when choosing an evaluation metric for a classifier, such as precision-recall, the receiver operator characteristic (ROC), and the F1 measure. However, because we are dealing with a highly imbalanced problem, we choose precision-recall as our major metrics following the argument in Ref. [16]. Specifically, the precision is the percentage of predicted positive samples that are truly positive, while recall is the percentage of truly positive samples that are also predicted to be positive.

In the context of our problem, the precision represents the percentage of predicted complaint bursts that will truly happen, while recall equals the percentage of all incoming complaint bursts that will be detected beforehand. Because positive predictions will lead to physical efforts to diagnose and fix malfunctioning BSs, a low precision will incur a substantial cost in real-life operation. Therefore, we favor precision over recall, and only present the system performance at a high-precision low-recall regime.

2) *Feature Time Windows:* The choice of feature time window determines how much information the machine-learning algorithm can be obtained from the raw data. We tested the proposed system under different feature time-window configurations to study the influence of the total window length, window granularity, and lead time on system performance.

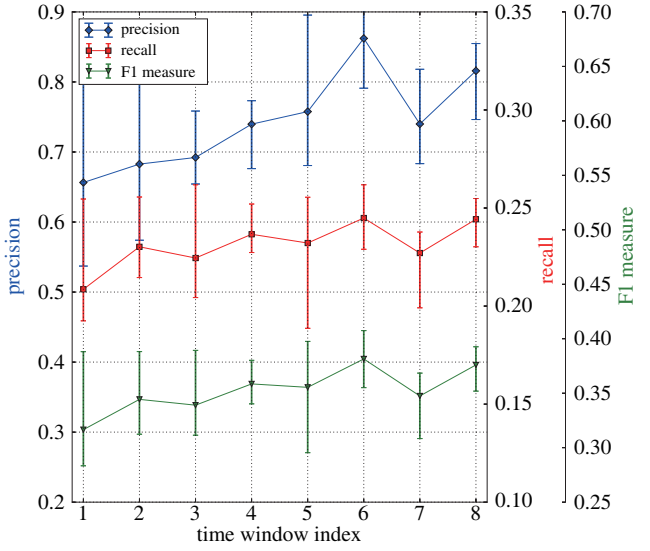
The total length of the feature time window determines how far the model can look back into history. The system performance under feature window configurations for different total lengths is shown in Fig. 4. Each point in the figure represents the median of 10 independent runs, and the corresponding error bar indicates the 10% and 90% percentile. The window configurations that are used are also listed in the table below. To isolate the influence of the total window length, we fix 4 time windows and a 3-h lead time in all configurations. As can be seen, both the precision and recall increase with the total length of the time window. This is intuitive because a longer window length will reveal more historic information. However, it should be noted that $tw8$ exhibits greater precision than $tw7$ at the cost of a larger recall, which indicates that the gain from the longer time window is saturating beyond around 100 h.

Another function of the feature time windows is to segment the history to reveal temporary details. The influence of the



window	detail	length
*tw*1	(3, 3), (4, 4), (5, 5), (6, 6)	4
*tw*2	(3, 3), (4, 4), (5, 7), (8, 12)	10
*tw*3	(3, 3), (4, 6), (7, 12), (13, 24)	22
*tw*4	(3, 3), (4, 12), (13, 24), (25, 36)	34
*tw*5	(3, 3), (4, 12), (13, 24), (25, 48)	46
*tw*6	(3, 3), (4, 12), (13, 24), (25, 72)	70
*tw*7	(3, 3), (4, 12), (13, 24), (25, 96)	94
*tw*8	(3, 3), (4, 12), (13, 24), (25, 120)	118

Figure 4 Precision and recall values on test sets with increasing length in feature time windows



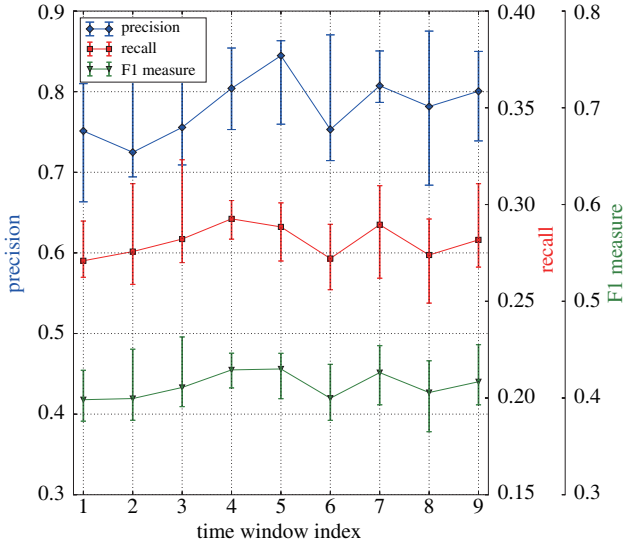
window	detail	number
*tw*1	(3, 3), (4, 96)	2
*tw*2	(3, 6), (7, 96)	2
*tw*3	(3, 3), (4, 12), (13, 96)	3
*tw*4	(3, 6), (7, 15), (16, 96)	3
*tw*5	(3, 3), (4, 12), (13, 24), (25, 96)	4
*tw*6	(3, 6), (7, 12), (13, 48), (49, 96)	4
*tw*7	(3,3), (4,12), (13,24), (25,72), (73,96)	5
*tw*8	(3,3), (4,12), (13,24), (25,48), (49,96)	5

Figure 5 Precision and recall on test set with different number of feature time windows

number of feature time windows is illustrated in Fig. 5. For all configurations, we set a lead time of 3 h and a total window length of 96 h. As can be seen in the table below, the greater the number of feature time windows, the finer will the total window length be segmented. The overall trend in Fig. 5 indicates that a finer segmentation (from *tw*1 to *tw*6) can help to improve the precision and recall. This is intuitive because a finer segmentation will reveal more temporal information. Nevertheless, we also note that over-segmentation (*tw*7 and *tw*8) will degrade the system performance. A possible explanation is that too large a number of feature time windows will greatly increase the dimensionality of the feature vector and dilute the training dataset, preventing machine-learning models from successfully generalizing the decision rule.

The prediction lead time is another important parameter. It is defined as the delay between the predicted event and the last feature time window. The lead time determines how far the predictor should look into the future, e.g., a 5-h lead time means that it predicts whether or not an event will happen 5 h later. A longer lead time will provide mobile operators with more time to diagnose equipment, and is therefore more favorable. The precision and recall values under varying lead times are shown in Fig. 6. Here, the prediction is for $s_t = 0$, and all configurations are simply time-shifted versions of *tw*1. As can be seen, the system performance tends to decrease as the lead time increases from 1 to 5, but it stops decreasing when we continue to increase the lead time. This result may appear counter-intuitive because it should be harder to predict further into the future. However, an analysis of the system's performance for a different BS will shed light on the unexpected result.

3) *Target Time Window:* The parameter target time window plays a pivotal role in our proposed system. As shown in Fig. 7, the system performance increases steadily and significantly with the length of the target time window. A straightforward explanation is that one complaint burst event can turn all target time windows that cover it into positive values. Thus, increasing the length of the target time window will result in more positive samples and help to alleviate the high skewness of raw events. A deeper insight is that human-generated complaint events are inherently uncertain in time, therefore the fuzziness introduced by longer target time windows can resolve such uncertainty. Note that a longer time window also makes the system overly pessimistic, and tends to exaggerate the situation. Consider the extreme case in which a year-long target window is likely to raise an alarm every hour. There-



window	detail	lead time
$tw1$	(1, 1), (2, 10), (11, 48), (49, 96)	1
$tw2$	(2, 2), (3, 11), (12, 49), (50, 97)	2
$tw3$	(3, 3), (4, 12), (13, 50), (51, 98)	3
$tw4$	(5, 5), (6, 14), (15, 52), (53, 100)	5
$tw5$	(7, 7), (8, 16), (17, 54), (55, 102)	7
$tw6$	(9, 9), (10, 18), (19, 56), (57, 104)	9
$tw7$	(12, 12), (13, 21), (22, 59), (60, 107)	12
$tw8$	(18, 18), (19, 27), (28, 65), (66, 113)	18
$tw9$	(24, 24), (25, 33), (34, 71), (72, 119)	24

Figure 6 Precision and recall on test set with different lead time

fore, it is more practical to set the target time window to a reasonably small length, e.g., 15 h.

4) *Resampling Ratio:* To deal with the imbalance of positive and negative samples, we re-sampled the training set before training the machine-learning model. Fig. 8 shows the precision and recall value as we increase the target p/n ratio (r_{PN}), which is the ratio between the number of positive and negative samples after resampling. As can be seen, the general trend is that a higher r_{PN} will increase the recall at the cost of lower precision. Intuitively, r_{PN} represents how we want to distort the original distribution: a higher value will emphasize the positive samples, making the classifier more aggressive, while lower values have the opposite effect, and makes the classifier more conservative. In addition, the increase in the recall is faster than the decrease in the precision. Therefore, it is more suitable to resample with a higher r_{PN} and then to weak the precision and recall values with the classifier threshold.

5) *Choice of Classifier:* We also experimented with three commonly used classifiers, i.e., LR, DTs, and RFs. Fig. 9 compares their precision and recall values under varying target time windows. We performed parameter searches for all three classifiers, and only the best parameter is shown. All of

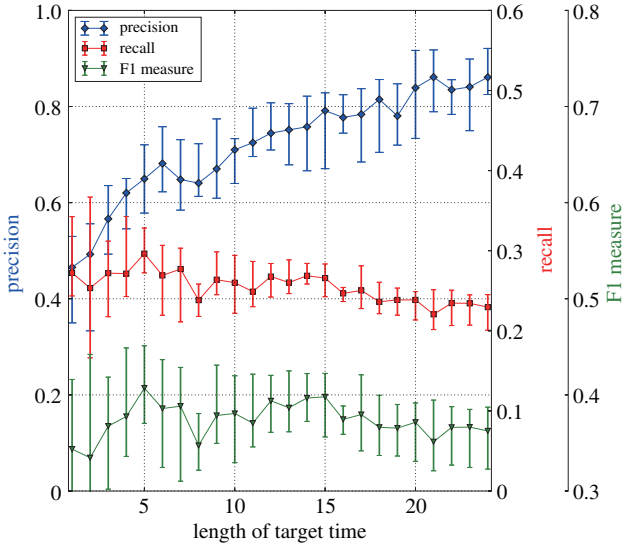


Figure 7 Precision and recall values under target time windows with different lengths

the classifiers used a decision threshold of 0.5. The RF has a clear advantage over the other two in terms of precision, while the recall values are comparable. Because we can always increase the recall by decreasing the precision, RF is generally the best option from among the three.

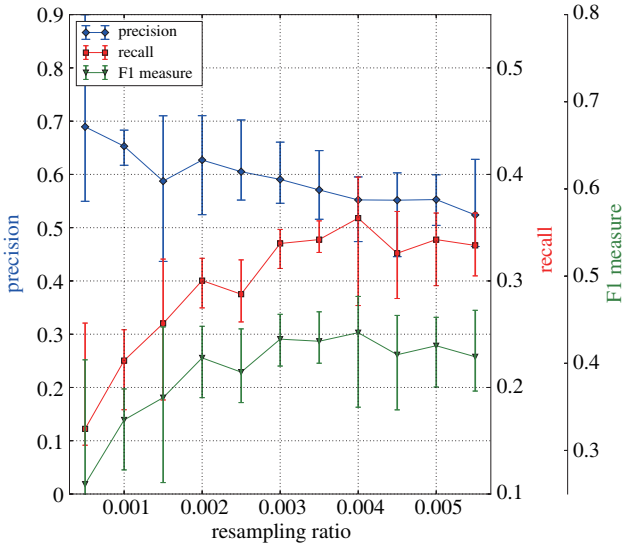


Figure 8 Effect of positive and negative event sampling ratio on prediction accuracy

V. Conclusion

In this paper, we presented PANDA, which is a user complaint prediction system for mobile access networks based on network monitoring data. PANDA can relate user complaints with network performance indicators, alarm reports, and other

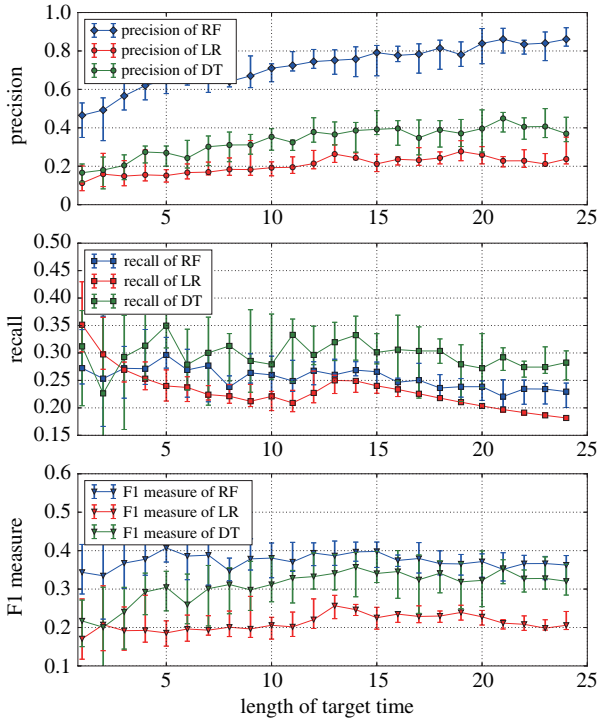


Figure 9 Precision and recall values of three different classifiers under varying target time windows

data in a data-driven fashion, and it can predict the occurrence of complaint events in a fine-grained spatial area within a certain time window. PANDA was evaluated on a real dataset collected from a major Chinese mobile network, where complaint burst events account for only about 0.3% of all recorded events. Results show that our proposed system can detect 30% of events in complaint bursts 3 h ahead with more than 80% precision. This will realize a corresponding degree of QoE improvement if all detected complaint events can be mitigated in advance by employing proper network maintenance.

In addition to being applicable for complaint prediction issues, it may be applied for many other similar issues that attempt to use one of two datasets that are interrelated in order to predict another dataset, our proposed system may probably work well. As future work, we aim to identify more suitable scenes to which our proposed system can be applied.

References

[1] N. Feamster, D. G. Andersen, H. Balakrishnan, et al. Measuring the effects of internet path faults on reactive routing [J]. ACM SIGMETRICS Performance Evaluation Review, 2003, 31(1): 126-137.

[2] L. Ma, T. He, A. Swami, et al. Node failure localization via network tomography [C]//ACM SIGCOMM Conference on Internet Measurement Conference, Vancouver, 2014: 195-208.

[3] X. Cheng, L. Fang, L. Yang, et al. Mobile big data: The fuel for data-driven wireless [J]. IEEE Internet of Things Journal, 2017, 4(5): 1489-1516.

[4] D. R. Choffnes, F. E. Bustamante, Z. Ge. Crowdsourcing service-level network event detection [C]//ACM SIGCOMM, New Delhi, 2010.

[5] A. Gerber, J. Pang, O. Spatscheck, et al. Speed testing without speed tests: estimating achievable download speed from passive measurements [C]//ACM SIGCOMM conference on Internet measurement, Melbourne, 2010: 424-430.

[6] H. Yan, A. Flavel, Z. Ge, et al. Argus: End-to-end service anomaly detection and localization from an isp's point of view [C]//IEEE INFOCOM'12, Orlando, 2012: 2756-2760.

[7] X. Cheng, L. Fang, X. Hong, et al. Exploiting mobile big data: sources, features, and applications [J]. IEEE Network Magazine, 2017, 31(1): 72-79.

[8] P. V. Klaine, M. A. Imran, O. Onireti, et al. A survey of machine learning techniques applied to self organizing cellular networks [J]. IEEE Communications Surveys and Tutorials, 2017, 19(4): 2392-2431.

[9] J. Friedman, T. Hastie, R. Tibshirani. The elements of statistical learning [M]. Springer, Berlin, 2009.

[10] L. Xiao, Y. Li, G. Han, et al. A secure mobile crowd sensing game with deep reinforcement learning [J]. IEEE Transactions on Information Forensics and Security, 2018, 13(1): 35-47.

[11] J. B. Caberera, B. Ravichandran, R. K. Mehra. Statistical traffic modeling for network intrusion detection [C]//8th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, San Francisco, 2000: 466-473.

[12] Z. Zhang, J. Li, C. Manikopoulos, et al. Hide: A hierarchical network intrusion detection system using statistical preprocessing and neural network classification [C]//IEEE Workshop on Information Assurance and Security, West Point, 2001: 85-90.

[13] C. Manikopoulos, S. Papavassiliou. Network intrusion and fault detection: A statistical anomaly approach [J]. IEEE Communications Magazine, 2002, 40(10): 76-82.

[14] B. Nguyen, Z. Ge, J. Van der Merwe, et al. Absence: Usage-based failure detection in mobile networks [C]//21st Annual International Conference on Mobile Computing and Networking, Paris, 2015: 464-476.

[15] S. Sun, X. Wei, L. Wang, et al. Association analysis and prediction for IPTV service data and user's QoE [C]//International Conference on Wireless Communications & Signal Processing, Nanjing, 2015: 1-5.

[16] J. Davis, M. Goadrich. The relationship between precision-recall and ROC curves [C]//23rd International Conference on Machine Learning, Pittsburgh, PA, 2006: 233-240.

About the Authors

Huimin Pan received his B.S. degree in Physics and his M.S. degree in electronic engineering from Tsinghua University, China, in 2013 and 2017, respectively. He is currently a researcher at the Cinda Jinyu (Shanghai) Investment Management Co., Ltd., and his research interests include machine learning, IoT, and big data analysis.

Sheng Zhou received his B.E. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2005 and 2011, respectively. From January to June 2010, he was a visiting student at the Wireless System Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA, USA. From November 2014 to January 2015, he was a visiting researcher in the Central Research Lab of Hitachi Ltd., Japan. He is currently an associate professor in the Department of Electronic Engineering, Tsinghua University. His research interests include cross-layer design for multiple antenna systems, mobile edge computing, and green wireless communications.

Yunjian Jia [corresponding author] received his B.S. degree from Nankai University, China, and his M.E. and Ph.D. degrees in Engineering from Osaka University, Japan, in 1999, 2003, and 2006, respectively. From 2006 to 2012, he was a researcher with the Central Research Laboratory, Hitachi, Ltd., where he engaged in research and development on wireless networks, and contributed to LTE and LTE-Advanced standardization in 3GPP. He is now a professor at the College of Communication Engineering, Chongqing University, Chongqing, China. He is the author of more than 80 published papers, and the inventor of more than 30 granted patents. His research interests include future radio access technologies, mobile networks, and IoT.

Dr. Jia has won several prizes from industry and academia including the IEEE Vehicular Technology Society Young Researcher Encouragement Award, the IEICE Paper Award, the APCC2017 Best Paper Award, the China Industry-University-Research Institute Collaboration Innovation Award, the Yokosuka Research Park R&D Committee YRP Award, and the Top 50 Young Inventors of Hitachi. Moreover, he was a research fellowship award recipient of both International Communication Foundation and the Telecommunications Advancement Foundation Japan.

Zhisheng Niu graduated from Northern Jiaotong University (currently Beijing Jiaotong University), China, in 1985, and received his M.E. and D.E. degrees from Toyohashi University of Technology, Japan, in 1989 and 1992, respectively. During 1992-1994, he worked for Fujitsu Laboratories Ltd., Japan, and in 1994 joined with Tsinghua University, Beijing, China, where he is now a professor in the Department of Electronic Engineering. He was a visiting Researcher at the National Institute of Information and Communication Technologies (NICT), Japan (10/1995-02/1996), Hitachi Central Research Laboratory, Japan (02/1997-02/1998), Saga University, Japan (01/2001-02/2001), Polytechnic University of New York, USA (01/2002-02/2002), University of Hamburg, Germany (09/2014-10/2014), and University of Southern California, USA (11/2014-12/2014). His major research interests include queueing theory, traffic engineering, mobile Internet, radio resource management of wireless networks, and green communication and networks.

Dr. Niu has served as the Chair of Emerging Technologies Committee (2014-2015), Director for Conference Publications (2010-2011), and Director for Asia-Pacific Board (2008-2009) of IEEE Communication Society, Councilor of IEICE-Japan (2009-2011), and a member of the IEEE Teaching Award Committee (2014-2015) and IEICE Communication Society Fellow Evaluation Committee (2013-2014). He has also served as associate editor-in-chief of IEEE/CIC joint publication China Communications (2012-2016), and editor of IEEE Wireless Communication (2009-2013), editor of Wireless Networks (2005-2009). He is currently serving as an area editor of IEEE Trans. Green Commun. & Networks, and is a Director for Online Content of IEEE ComSoc (2018-2019).

Dr. Niu has published 100+ journal and 200+ conference papers in IEEE and IEICE publications and co-received the Best Paper Awards from the 13th, 15th and 19th Asia-Pacific Conference on Communication (APCC) in 2007, 2009, and 2013, respectively, International Conference on Wireless Communications and Signal Processing (WCSP13), and the Best Student Paper Award from the 25th International Teletraffic Congress (ITC25). He received the Outstanding Young Researcher Award from the Natural Science Foundation of China in 2009 and the Best Paper Award from the IEEE Communication Society Asia-Pacific Board in 2013. He was also selected as a distinguished lecturer of the IEEE Communication Society (2012-2015) as well as IEEE Vehicular Technologies Society (2014-2016). He is a fellow of both IEEE and IEICE.

Meng Zheng received his B.S. and M.Sc. degrees in information sciences from Beijing Institute of Technology in 2008 and 2010, respectively. He is currently a senior researcher at the Hitachi (China) Research & Development Corporation, Beijing. He worked as a 3GPP standards expert from 2011 to 2014. His research interests include wireless networking architecture for 5G, and big data analysis in the telecommunication industry.

Lu Geng received her B.S. and M.Sc. degrees from the Beijing University of Posts and Telecommunications, in 2002 and 2005, respectively. She is currently a chief researcher at the Hitachi (China) Research & Development Corporation, Beijing. She worked as a 3GPP standards expert from 2009 to 2014. Her research interests include wireless networking technologies, IoT, and big data analysis in the telecommunication industry.