Experience: Towards Automated Customer Issue Resolution in Cellular Networks

Amit Sheoran
Purdue University
asheoran@purdue.edu

Sonia Fahmy
Purdue University
fahmy@purdue.edu

Matthew Osinski
AT&T Labs Research
mosinski@research.att.com

Chunyi Peng
Purdue University
chunyi@purdue.edu

Bruno Ribeiro
Purdue University
ribeiro@cs.purdue.edu

Jia Wang
AT&T Labs Research
jiawang@research.att.com

ABSTRACT

Cellular service carriers often employ *reactive* strategies to assist customers who experience non-outage related individual service degradation issues (e.g., service performance degradations that do not impact customers at scale and are likely caused by network provisioning issues for individual devices). Customers need to contact customer care to request assistance before these issues are resolved. This paper presents our experience with PACE (ProActive customer CarE), a novel, *proactive* system that monitors, troubleshoots and resolves individual service issues, without having to rely on customers to first contact customer care for assistance. PACE seeks to improve customer experience and care operation efficiency by *automatically* detecting individual (non-outage related) service issues, prioritizing repair actions by predicting customers who are likely to contact care to report their issues, and proactively triggering actions to resolve these issues. We develop three machine learning-based prediction models, and implement a fully automated system that integrates these prediction models and takes resolution actions *for individual customers*. We conduct a large-scale trace-driven evaluation using real-world data collected from a major cellular carrier in the US, and demonstrate that PACE is able to predict customers who are likely to contact care due to non-outage related individual service issues with high accuracy. We further deploy PACE into this cellular carrier network. Our field trial results show that PACE is effective in proactively resolving non-outage related individual customer service issues, improving customer experience, and reducing the need for customers to report their service issues.

CCS CONCEPTS

• **Networks** → **Mobile networks**; • **Computing methodologies** → *Machine learning*; • **Applied computing** → *Forecasting*;

KEYWORDS

Cellular Networks; Decision Trees

MobiCom '20, September 21–25, 2020, London, United Kingdom

ACM ISBN 978-1-4503-7085-1/20/09.
https://doi.org/10.1145/3372224.3419203

ACM Reference Format:
Amit Sheoran, Sonia Fahmy, Matthew Osinski, Chunyi Peng, Bruno Ribeiro, and Jia Wang. 2020. Experience: Towards Automated Customer Issue Resolution in Cellular Networks. In *The 26th Annual International Conference on Mobile Computing and Networking (MobiCom '20), September 21–25, 2020, London, United Kingdom.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3372224.3419203

1 INTRODUCTION

Cellular service providers (carriers) are constantly pushing the boundaries to deliver a positive, meaningful and unique experience for each customer (user or subscriber). The vast majority of cellular network outages that impact customers at scale are proactively detected and resolved without waiting for customers to report them. However, service degradation caused by individual customer provisioning and device configuration errors still largely rely on customers to make the the first move. To meet customer needs, an unprecedented number of traditional and digital channels have been made available, as customer support can be provided over the phone, through social media, and by online virtual assistants. While these omni-channel strategies have transformed how customer experience is managed, many of these strategies are largely reactive . Carriers tend to investigate and resolve these non-outage related individual customer service performance issues only after the customer initiates a trouble request.

In this experience paper, we report on our first attempt to improve each individual customer's experience and increase cellular service operation efficiency by shifting from a *reactive* strategy to a *proactive* strategy when dealing with individual customer issues. We develop PACE (ProActive customer CarE), a novel proactive framework that automatically detects non-outage related service issues that impact individual customers' experience, and predicts a future customer care interaction as a result of these service issues to prioritize resolution actions. PACE further triggers resolution actions to remedy the detected issues before customers contact support agents. This not only improves customer experience by minimizing the impact of service issues, but also increases operation efficiency of cellular service providers by reducing the number of customer care contacts and the subsequent investigation and mitigation process.

We present our experience with PACE through a field trial in a major cellular carrier network in the US. We focus on non-outage related technical issues (e.g., user equipment (UE) and service related issues) that impact individual customer experience (e.g., network

connectivity issues caused by individual customer network provisioning or device configuration errors), as other customer issues such as cellular network outages, cellular service plan upgrades and billing events are out of our scope. We seek a deeper understanding of these technical issues reported by customers, and utilize machine learning techniques to identify network and service performance degradation signatures that lead customers to contact customer support or customer service (referred to as "care" in the remainder of this paper). Although prior research has considered the problems of automating fault diagnosis (e.g., [11]) and mining customer care contact logs to classify and infer problems and events (e.g., [7, 10, 13, 16, 20]), to the best of our knowledge, no work has attempted to correlate network logs and customer care logs to automatically take resolution actions for *individual customers*.

Our work aims to answer the following questions: (1) What types of device and network issues are likely to cause service performance degradation that drives customers to contact care and report their issues? (2) Which actions should be invoked on customer devices and network servers to resolve customer issues? and (3) How effective are proactive resolution actions in terms of improving customer experience and increasing service provider operation efficiency?

The problem of identifying which individual customers are impacted by service issues is complicated by a number of factors. *First*, while our data sources are massive, some do not have the granularity or latency desired to compose a complete picture of individual service issues across the end-to-end service path. *Second*, not all customers report service troubles. Only a small fraction of customers contact customer care to report service degradations. Furthermore, not all customers report technical issues at the moment they occur and instead wait a period of time that is highly variable. *Third*, the service issues that customers report to care can be ambiguous or inaccurate. The issue types recorded by customer care agents can be highly subjective, and it is difficult to attribute them to specific root causes. Expanded service offerings, new devices on the market, and multiple resolution actions exacerbate the problem.

This paper makes the following contributions: (1) We propose and describe our experience with PACE, an automated framework to identify individual customers experiencing service degradation due to technical issues, and resolve their issues by proactively invoking resolution actions. (2) We analyze data sources available to cellular service providers to understand customer behavior. (3) We develop novel machine learning-based models to predict individual customers who are likely to contact care due to non-outage related service issues using a combination of customer-perceived and network-observed metrics. (4) We conduct trace-driven evaluation of PACE based on large-scale real-world data collected from a major cellular service provider. (5) We deploy PACE into the major cellular service provider network and launch a field trial. Our field trial results show that PACE is effective in proactively resolving non-outage related individual service issues, improving customer experience, and reducing customer care contacts.

Roadmap. §2 defines our datasets and gives some background. §3 discusses the challenges introduced by dataset characteristics and content. §4 explains our framework, PACE, and machine learning-based models. §5 gives our experimental results and §6 describes our field trial. §7 summarizes related work. §8 discusses limitations and extensions to our work. §9 concludes the paper.

This work does not raise any ethical issues. We do not use any customer demographic or geographic location information in this paper.

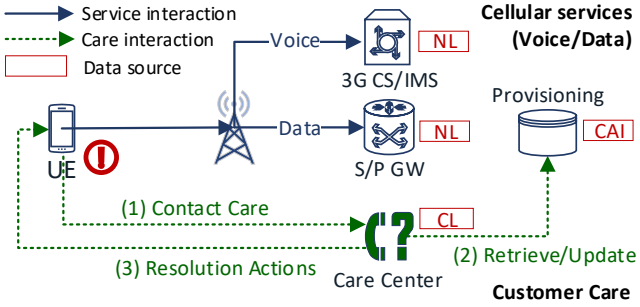


Figure 1: Basic cellular service and customer care flows and data collection interfaces.

2 BACKGROUND AND DATASETS

We begin with necessary background on cellular services and customer care provided by network operators, and then introduce the data sources.

2.1 Cellular Services and Customer Care

Figure 1 presents an overview to support basic cellular services (top), as well as the typical work flow used to resolve customer issues (bottom). Cellular network operators provide a plethora of services to their customers by multiplexing customer-generated traffic between user equipment (UE, here, mobile phones) to their targeted endpoints (e.g., servers/hosts at the external Internet or mobile phones) through their network infrastructure. The network infrastructure consists of cascading network functions (NFs) for radio access and core network access. Today, most operators support several co-existing technologies (e.g., 2G/3G/4G/5G). In this work, we focus on 3G/4G and beyond which is the state-of-practice across the United States, and we focus on two basic services within those domains: *voice* and *data*. Specifically, 3G/4G radio access is offered by base stations, i.e., eNodeB. In the core network, data service is provided by Serving/Packet Gateway (S/PGW) in 4G, whereas voice call service is provided by voice-over-LTE (VoLTE) which traverse 4G NFs at IP Multimedia Subsystems (IMS), or circuit-switched fallback (CSFB) that traverses the legacy 3G circuit-switched (CS) network [1].

Issues that impact a specific user's session across the control or user planes may occur anywhere across the network topology, and are captured via network fault and performance management techniques. While the vast majority of cellular network outages that impact customers at scale are proactively detected and resolved without waiting for customers to report them, customer care organizations employ *reactive* strategies when dealing with individual customers, as customers usually have to initiate a trouble report to generate an investigation for many non-outage related individual service performance issues. The bottom part of Figure 1

shows a typical customer care investigation with three steps. First, the customer contacts the care center and is directed through an automated interactive voice response system to the appropriate first-tier team. Second, according to the issue type (e.g., billing, administration, technical), the first-tier team performs a number of routine troubleshooting and resolution actions, such as checking customer account and billing status, service provisioning status, known outages and hardware/software issues. If the issue is not easily diagnosed and resolved, it is escalated to the second-tier technical team for further investigation of network and service logs, along with performing additional resolution actions. All the actions are automatically recorded in this process. The customer care agent who is responsible for this customer-reported issue also provides a written summary, including the troubleshooting steps (resolution actions) taken and their results.

Customer care agents handle a large variety of individual customer issues, which are roughly divided into two categories: technical and non-technical. A technical issue means that it is likely a result of provisioning/configuration issues in the network/device, for example, "*Unable to make/receive voice calls,*" "*Unable to connect to data services,*" and "*Cellular Data Connectivity.*" Non-technical issues are related to routine customer engagement or information inquiries, such as inquires about new services or plans, activation/deactivation, billing-related inquires, and device/hotspot setup. In general, technical issues are difficult to resolve and may impact customer experience, so we focus on the automated resolution of *technical issues* in this paper.

2.2 Data Sources

Our aim is to replace the reactive strategy with a *proactive* one that automatically detects and resolves non-outage related service issues that impact only individual customers, in order to reduce the resolution time and improve customer experience. We detect performance degradation issues faced by a customer through leveraging network, service, and customer care logs collected by the cellular service provider.

As illustrated in Figure 1, we have several data sources collected and aggregated from existing networking interfaces. The main data sources used in this work can be summarized as follows.

1. Care Logs (CL). Care logs record interactions between customers and care agents, and include (but are not limited to): (1) user ID (UID), (2) timestamp, (3) care contact channel including online chats, phone calls, and store walk-ins, (4) issue type which is manually provided by the interacting care agents, (5) description, which is in free text format added by the agent, and (6) sequence of actions taken for troubleshooting and resolution of customer issues. Note that the care logs are collected after obtaining customers' permission. In this paper, we only consider customer feedback received from care calls, and we do not use customer feedback received from other channels such as online chat, store walk-ins, or user posts on social media sites. Our analysis only considers customers who have called customer care to report service quality degradation.

2. Customer Account Information (CAI): Customer account information contains information regarding the customer service subscription such as UID , the device manufacture and model, hardware and software version, activation time, and last update to the account.

Table 1: Attributes in the VCR dataset.

Attribute	Description
UID	User identifier (anonymized)
Start Time	Time when the Charging Collection Function (CCF) started the session
End Time	Time when the CCF terminated the session
Cause for Record Closing	Reason for the release of the session (0) for successful sessions
Status (if applicable)	Abnormal status information of the session SIP (4XX/5XX) code (Blocked/Dropped)

Table 2: Attributes in the DCR dataset.

Attribute	Description
UID	User identifier (anonymized)
Start Time	Time when the PDN session starts
End Time	Time when the PDN session ends
CFT Code	Cause for termination (CFT) for a PDN session
APN ID	APN name of failed PDN session

3. Network Logs (NL). The network logs contain information regarding how a customer device uses each data/voice service over the network. It consists of two datasets:

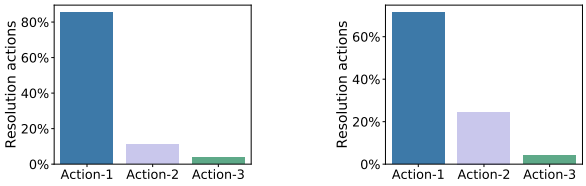
3a. Voice Call Records (VCR). This records information for each voice call. Data is collected by the IMS for VoLTE calls, and by the 3G CS network elements for CSFB calls. Table 1 lists its main attributes (additional attributes can be found in the standards [1, 3]). The dataset covers both successful voice calls and failures. In case of failures, additional information are recorded to capture the status and cause of record closing.

3b. Data Connection Records (DCR). This records information for each data service. It is collected from the gateways for each packet data network connectivity (PDN) session organized by its Access Point Name (APN). Table 2 lists the main attributes [2, 3]. In case of packet-switched (PS) calls (VoLTE), a predesignated APN is used to tunnel a voice call from the PDN network to IMS, and the sessions are recorded in the VCR dataset. If a session fails, the cause of termination is recorded as a CFT code.

3 OBSERVATIONS AND CHALLENGES

Our approach to automating customer issue resolution applies machine learning techniques to proactively identify individual customers who are experiencing a degraded service experience which is non-outage related, and predict if they will contact care in the near future. We use customer care contact as a measure of severity of impact and prioritize automated repair actions accordingly. This problem can be modeled as a classification problem using the features extracted from datasets collected by the service provider. If a customer is predicted to call care, we take proactive action to resolve the issue, which will reduce the severity and duration of customer service quality degradation, and eliminate the need for the customer to contact care.

Figure 2: Time after which customers contact care.



(a) Actions to resolve 'Cannot Make or Receive Calls.'

(b) Actions to resolve 'Cellular Data Connectivity.'

Figure 3: Distribution of resolution actions taken by care agents.

3.1 Challenges

While determining if an individual customer is experiencing service quality degradation sounds simple, there are several technical challenges that are inherent in the datasets used in the study.

Data scale and quality. The datasets described in §2.2 were collected from a major cellular service provider in the US over several months. Developing models that generate actionable results for individual customers requires data sources that provide UE granular information with a low to medium latency. Furthermore, an analysis of the NL datasets (VCR + DCR) for failures requires significant computational resources due to the massive volume of the NL datasets (e.g., over 100 billion entries per day).

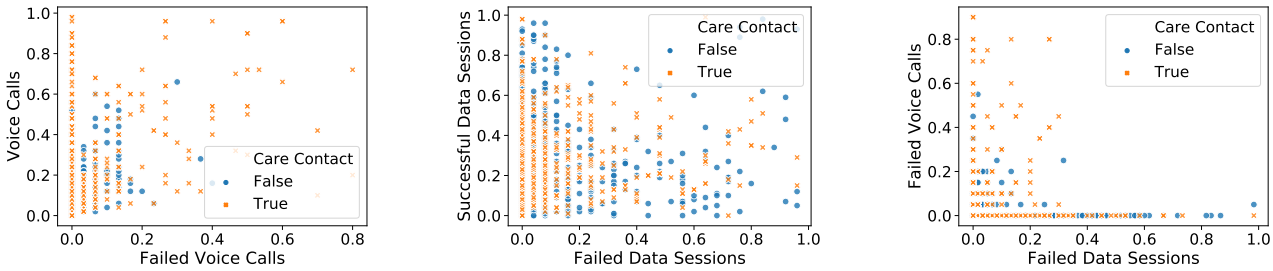
Not all candidate data sources meet these requirements, which limits the ability to capture the metrics from the end-to-end service path of customers' cellular service. To gain a better understanding on this data limitation, we correlated the CL and NL datasets, and computed the percentage of customer-reported issues in the CL dataset that we can observe from the NL dataset.

Behavioral aspects in issue reporting. By correlating the CL and NL datasets, we found that only a very small percentage of customers who experience some kind of service degradation (e.g., a voice call drop) contact care. In addition, customers who experience performance-related issues do not necessarily have a higher probability of contacting care and reporting their issues. Figure 4a shows the normalized count of voice calls made by customers during a time period of two weeks and the normalized count of voice call failures experienced by them. As seen from Figure 4a, a large percentage of customers who call care experience few voice call failures. More precisely, >50% of customers who call care experience two or fewer failed calls. A similar pattern is seen in Figure 4b which compares the normalized count of data sessions initiated by customers and the normalized count of data connectivity failure events experienced by them. We found that customers who contacted care do not always encounter a large number of data connectivity failures; in fact, a large number of customers who contact care encounter few (if any) data connectivity failures (Figure 4b). These observations imply that threshold-based prediction models, e.g., [10], cannot be used to predict customer behavior and to prioritize our proactive resolution actions.

Customers who contact care. In addition, differing customer attitudes [8, 12] impact when and how often a customer contacts care. Figure 2 shows the distribution of the time difference between the last observed failure event in the NL dataset and the time when the customer contacts care to report technical issues. We found that, among all customers who contact care, only ~18% of customers contact care within 24 hours of experiencing an issue, and ~27% of customers contact care two days after experiencing an issue. The duration after which a customer contacts care depends on when the performance issue is encountered (e.g., the day of the week and time of day) and the failure patterns/signatures. In addition, different customers likely have different tolerance levels to performance degradation. For example, most customers may choose to ignore temporary performance degradation unless the issue is persistent or chronic, while some customers tend to contact care every time they spot a performance degradation. Predicting individual customers who are likely to contact care due to technical issues therefore requires understanding the relation between network-observed metrics (Quality of Service (QoS)) and customer-perceived experience. Since customer experience is affected by individual customer behavior and attitude (especially for non-outage related service degradation that impacts only individual customers), developing models tailored to improve individual customer experience is a challenging task.

Issue classification and resolution fidelity. When a customer does report a service issue, the issue types recorded can be highly subjective and sometime ambiguous. While service degradation can occur across the end-to-end service path, individual customers are seldom aware of the root causes. Given the complexity of the network, along with emerging technologies and changes in the device ecosystem, the actual categories of errors reported to care by customers are very broad. For example, some of the most commonly reported issues include "Unable to make voice call" and "Unable to connect to internet."

In general, care agents are trained to follow a predefined protocol during their interactions with customers. Due to the subjectivity and ambiguity of issues reported by customers, we observe inconsistency in diagnosis and resolution actions performed by care agents. Figure 3 shows the distribution of three resolution actions taken by care agents to resolve issues in the category "Cannot Make or Receive Calls" and "Cellular Data Connectivity." While we omit the specific resolution action name for confidentiality, these resolution



(a) Relationship between successful and failed voice calls.

(b) Relationship between successful and failed data sessions.

(c) Relationship between voice call failures and data connectivity failures.

Figure 4: Analysis of voice/data usage and failures for a sample population of customers in the training data.

actions include triggering a network initiated detach procedure [2], forcing the customer device to re-initiate the radio connection, and flushing the authentication vectors from the Home Subscriber Server (HSS).

Correlating network events to customer experience. The NL datasets contain many events and alerts. There are several reasons why it is difficult to map these events and alerts to a degraded customer experience. First, the end-to-end service path of a cellular service request usually contains multiple network elements, some of which may be external to the cellular carrier network. Example external elements include a callee who is a customer of a different cellular carrier, and 3rd party message application servers. Second, mobility imposes dynamics on the serving cell sites for a given customer. Third, cellular service providers deploy and manage cell sites in such a way that customers (and their traffic) can be migrated from a degraded cell site to its neighboring cell sites without impacting customer experience. Finally, the complex protocol stacks and error codes used by cellular equipment vendors and service providers make the correlation between network events/alerts and customer experience very challenging. As an example, consider the SIP [18] response code recorded in Table 1. We found that a large number of failed messages contain the error code `408 Request Timeout`, which is issued by a SIP-supporting network function when the server is unable to produce a response within a suitable period of time. It can be issued if the server is unable determine the location of the customer or due to other internal timeouts.

3.2 Design Guidelines

The NL dataset includes several attributes that are not useful in predicting user behavior, such as error codes and network elements involved in processing cellular traffic. We use the following insights to select features used in the design of our prediction models (§4.3). *First*, most customers only experience either voice or data connectivity failures, but not both. This can be seen from Figure 4c, which shows the normalized count of failed calls versus the normalized count of failed data sessions experienced by customers. *Second*, customers have lower tolerance for voice call failures and therefore customers experiencing a higher number of voice call failures are more likely to report issues to care (Figure 4a). *Third*, customers who make a large number of voice calls are more likely to contact care and report issues (Figure 4a). In the data shown in Figure 4a, ~68% of customers in the top 10 percentile of voice call usage reported an issue to customer care. Therefore, we believe that customers who make a large number of voice calls and experience a greater number of failure events are more likely to contact care, which results in a higher percentage of voice call-related issues reported to care. Our model to predict customer care contact behavior should thus focus on customers who (a) make a high number of voice calls, and (b) experience a higher number of voice call failures.

4 THE PACE FRAMEWORK

4.1 Overview

PACE is a framework for automated monitoring, detection, and resolution of individual customer issues related to service performance degradation that are not caused by network outages. As depicted in Figure 5, PACE comprises two phases: (1) an *offline* phase in which historical data is used to train prediction models, and (2) an *online* phase in which the predictions made by the models are used as triggers to take resolution actions to repair issues impacting individual customer devices.

The offline phase consists of three steps. *First*, the three input datasets (CAI, CL, NL) are preprocessed. The main goal of this data preprocessing is to remove data related to devices or conditions that are out of scope of our study. *Second*, feature extraction is applied on each input dataset. *Third*, machine learning models are trained for predicting individual customers who are likely to contact care and report service issues based on the extracted features.

In the online phase, the prediction model will take real-time network logs as input, detect failure signatures, and predict the customers who are likely to contact care due to service issues. This allow us to prioritize proactive actions that need to be performed to resolve individual service issues based on their impacts. A predefined set of resolution actions is then triggered to resolve these issues. PACE does not currently predict the resolution actions that must be triggered to resolve service quality degradation. As seen from Figure 3, we observe significant variations in the actions taken by care agents to resolve a given customer issue, which makes predicting the next-best-action to resolve a service quality degradation a challenging problem. Service performance metrics are monitored

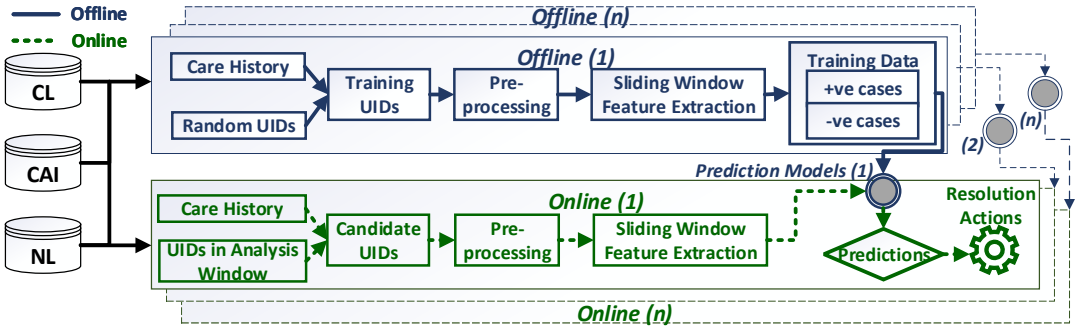


Figure 5: PACE framework design. Multiple instances of Offline and Online phases can be deployed independently at a central or distributed site(s).

before and after the actions to assess the impact of resolution actions.

We note here that PACE can be deployed in either centralized or distributed fashions. The offline and online phases of PACE can be deployed independently in separate geographical locations (Figure 5).

4.2 Data Preprocessing

We preprocess datasets to filter out data in the following three cases. *First*, we remove data records that are collected during the time period when there were known data quality issues (e.g., missing data, bogus data). This is an important step because suspect data can have undesirable impact on the prediction model. Automatically and systematically cleaning data has always been a challenge and is itself an open research problem. In this work, we use a rule-based mechanism to identify and remove suspect data based on domain knowledge.

Second, we filter out all the data records related to known network outages in both the CL and NL datasets. Individual service degradation which are not attributed to network faults account for a significantly high percentage of customer reported issues in our data sets, and are a much harder problem to diagnose and resolve at scale, when compared to service degradation stemming from network outages. Our work primarily focuses on the non-outage related service degradation issues that impact individual customer experience (e.g., device misconfiguration, provisioning errors, device software/hardware issues).

Third, we discard all roaming mobile devices from our NL datasets. We define a customer mobile device to be a *roaming device* if the customer is not an authenticated customer based on the CAI data and we are not able to conduct any resolution actions on these devices.

4.3 Feature Extraction

Our feature extraction method generates usable feature vectors from raw data. From §2.2, we have two types of data sources: (a) Static data sources such as the CAI logs for which the information can only be changed by cellular Operations and Management (OAM), and (b) dynamic data sources such as the care and network logs which change as customers use services or contact care.

Table 3: Static Feature Vectors per UID

Feature	Description
Activation-Time	Account activation timestamp
device-type-apple	True/False
device-type-others	True/False

Table 4: Dynamic Feature Vectors per UID

Feature	Description
# Blocked-Calls	Number of calls blocked
# Dropped-Calls	Number of calls dropped
# Calls	Number of calls made
# Call Duration	Duration of calls made
# Data-Session (F)	Number of failed data sessions
# Data-Sessions (S)	Number of successful data sessions
# Care-calls (T)	Number of technical care contacts
# Care-calls (O)	Number of other care contacts

Table 3 shows static feature vectors we extract from the CAI dataset for each unique device. We use one-hot-encoding to convert descriptive features such as "Device Information" to binary values (e.g., "True" or "False"). Similarly, the dynamic records in the CL and NL datasets are not directly usable as features. We calculate key performance metrics (e.g., usage and failure) using a fixed time bin (e.g., one day or one hour) as dynamic feature vectors at per UE granularity as listed in Table 4.

Straightforward feature extraction may have the following shortcomings due to challenges discussed in §3. **(1) Imbalanced classes for training:** Figure 4a shows that only a small fraction of customers experience service issues, and only a small fraction of these customers contact care. This imbalance is typically solved either by under-sampling the majority class in a mini-batch or re-weighting its loss, but fewer samples of the minority class (customers who contacted care) will require accumulating training data over a longer duration for a statistically significant balanced training sample. **(2) Seasonality within data:** Customer care centers experience higher call volumes on certain days of the week. **(3) Temporal dependencies:** Once a customer contacts care, the same customer

is unlikely to contact care if the issue is resolved. In addition, customers are unlikely to contact care on consecutive days, as we see in the CL dataset. **(4) Reporting latency.** Over 80% of customers who contact care wait at least 24 hours after experiencing service issues (Figure 2). It is insufficient to simply use previous day data.

Sliding window design. To account for behavioral aspects of care contact, maximize utilization of samples of customers who contact care, and exploit care contact seasonality, we propose a sliding window design to create the training/test data for our models. Figure 6 shows how features are aggregated to create the training/test set using a sliding window of 7 time units, W = {T−7, T−6, T−5, T−4, T−3, T−2, T−1}. As shown in Figure 6, using inputs of 10 time units (T−9 to T−0), we create three slices of data, slice-1 using data from T−9 to T−3 using which predictions for T−2 are generated, slice-2 (T−8 to T−2) using which predictions for T−1 are generated, and so on.

This sliding window design allows us to (a) maximize the utilization of positive samples (shortcoming (1) above): Data for each time unit is used in multiple slices (n times with an n-unit sliding window), and (b) handle seasonality and temporal dependencies (shortcomings (2) and (3) above): We use care contacts made by a customer per time unit as an additional input to the models. Another solution for addressing seasonality within the data is to create a model for each day of the week (Monday through Sunday), and use these day-of-week models to predict the outcome of a specific day. We evaluated the performance of day-of-week models and found that there is no significant difference in the performance of these models versus the performance of models that use the sliding window algorithm. Therefore, we omit the discussion of day-of-week models in the rest of the paper. To address shortcoming (4) above (unpredictable reporting time), we use a different time window for predicting if a customer is likely to contact care, e.g., we predict if a customer is likely to contact care in the next $|R|$ time units as a result of service issues, where set R = {T−0, T+1, T+2, T+3, T+4}. The sliding window parameters $|W|$ and $|R|$ should be carefully selected to meet operation requirements. The values of $|W|$ and $|R|$ used by PACE are discussed in §5.

4.4 Problem Formulation

Let $\mathcal{U}$ be the stochastic process that generates our data $U_i = (U_{i,1}, \ldots, U_{i,n})$, which are the records of the i-th user from a starting time $t_0 = 1$ until an end time $t_{\text{end}} = n$, where $n > |W| + |R|$ is assumed constant. Let $N > 0$ be the number of users in our logs, then $\{U_i\}_{i=1}^{N}$ is the data used to construct our training dataset. The training dataset $\mathcal{D}_{\text{train}}$ is constructed through sliding windows of length $|W| + |R|$ over $\{U_i\}_{i=1}^{N}$, as depicted in Figure 6. More specifically, we consider the training dataset as $\mathcal{D}_{\text{train}} = \{((\mathbf{X}_i, \mathbf{Z}_i), \mathbf{Y}_i)\}_i$, where $\mathbf{X}_i \in \mathbb{R}^{|W| \times p_{\text{stat}}}$ is a matrix with p_{stat}-dimensional *static* features of each time unit in W, $\mathbf{Z}_i \in \mathbb{R}^{|W| \times p_{\text{dyn}}}$ is a matrix with p_{dyn}-dimensional *dynamic* features of each time unit in W, and $\mathbf{Y}_i \in \{0,1\}^{|R|}$ is a random variable vector containing the target label (whether or not someone will contact the call center at each of the next $|R|$ time units in the future).

We now define a function (feature generator) $\phi : \mathbb{R}^{|W| \times p_{\text{dyn}}} \to \mathbb{R}^{|W| \times p_{\text{dyn}}}$ that takes $\mathbf{Z}_i$ of user i as input and outputs a set of features in the same space. We then learn a classifier f that takes this feature matrix $\phi(\mathbf{Z}_i)$ and $\mathbf{X}_i$ and outputs $\hat{\mathbf{y}} \in [0,1]^{|R|}$, an estimate of the probability

$$P((\mathbf{Y}_i)_d = 1 | \mathbf{X}_i, \mathbf{Z}_i) \approx (\hat{\mathbf{y}}_i)_d = f(\phi(\mathbf{Z}_i), \mathbf{X}_i)_d \quad (1)$$

that user i will contact care and report an issue at the d-th time unit of a window of size $|R|$, $i = 1, \ldots, N$.

4.5 Ensemble Model Design

A straightforward approach for solving the problem described in §4.4 is to build a binary classification model based a combination of static and dynamic feature vectors. However, this simple binary classification may not work well (§5) due to the challenges discussed in §3. We therefore investigate different feature vector generation functions ϕ from Equation 1 designed to address these challenges.

Aggregated Feature Model (AFM): AFM uses a combination of features extracted from (a) Static features in Customer Account Information (Table 3), i.e., $\mathbf{X}_i$, and (b) Aggregated dynamic feature vectors extracted from the NL and CL datasets (Table 4), i.e., $\mathbf{Z}_i$. AFM defines the ϕ_{AFM} of Equation 1 as the identity function, i.e., $\phi_{\text{AFM}}(a) = a$, which yields

$$\widehat{\mathbf{y}}_i^{(AFM)} = f(\phi_{\text{AFM}}(\mathbf{Z}_i), \mathbf{X}_i), \quad (2)$$

where $\widehat{\mathbf{y}}^{(AFM)}$ is the classifier described in Equation 1.

We also analyzed the feature importance scores generated by XGBoost [23] for the AFM. The top five features in decreasing order of feature importance scores are : Activation-Time, Call-Duration {T-1,T-2,T-3}, #Calls {T-1,T-2,T-3}, #Failed-Calls {T-1,T-2,T-3}, #Data-Sessions {T-1,T-2}. In case of dynamic features where each feature consists of $|W|$ entries (one entry corresponding to each time unit in W), the value in the parenthesis { } shows the prefix of the time unit which had the highest feature importance score (sorted in decreasing order). The feature importance scores of AFM are consistent with the observations in §3.2.

Individual Variations Model (IVM): IVM is designed to leverage variations in individual usage/failure patterns. As an example, consider a customer (in a low coverage area) who experiences an average of n voice call failures/day. While this customer may not contact care if they continue to experience similar failures ratio per day, they are likely to contact care when the number of failures exceeds the average daily failures. The IVM model is designed to detect such variations in individual usage.

The static features used by IVM model are the same as the AFM, since there are no variations in the CAI datasets. Dynamic features of the IVM are created by subtracting the actual values in a given time unit from the mean value of the same feature, where the mean value of a feature is calculated using all instances of a given feature in the entire data $\mathcal{D}$. The dynamic features used by IVM can be described as

$$\phi_{\text{IVM}}(\mathbf{Z}_i) = (\bar{\mathbf{z}}_i - \mathbf{Z}_i, \ldots, \bar{\mathbf{z}}_i - \mathbf{Z}_i), \quad (3)$$

where $\bar{\mathbf{z}}_i$ is a vector whose m-th component is the row-average (time average) of feature (column) m in matrix $\mathbf{Z}_i$.

The output of the classifier in Equation 1, using the features created by ϕ_{IVM}, is then

$$\widehat{\mathbf{y}}_i^{(IVM)} = f(\phi_{\text{IVM}}(\mathbf{Z}_i), \mathbf{X}_i). \quad (4)$$

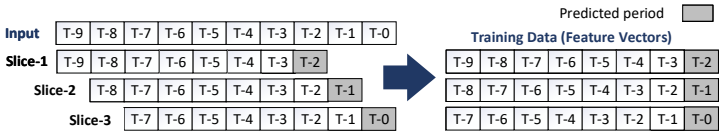


Figure 6: Sliding window design for feature vectors.

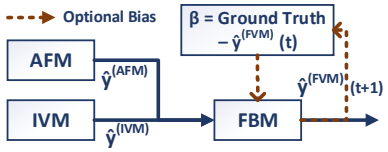


Figure 7: Control loop of the feedback model (FBM).

The feature importance score of IVM is similar to that of AFM and is therefore omitted for brevity.

Feedback Model (FBM): Obtaining a model with low false positives is one challenge in developing a framework for automated proactive customer issue resolution. We will show in §5 that both AFM and IVM have low precision and therefore result in high false positives. We would like to argue that, while overall model accuracy is important, minimizing false positives is critical in reducing unnecessary actions and any unwanted impacts on network and customer devices. We therefore design an ensemble model that uses the inputs from AFM and IVM to minimize the prediction errors in PACE.

Figure 7 depicts the design of FBM. The model uses the probability values generated by AFM and IVM, i.e., $\widehat{\mathrm{y}}^{(AFM)}$ and $\widehat{\mathrm{y}}^{(IVM)}$ from Equation 2 and Equation 4, respectively, as inputs along with a bias variable. Recall from Figure 6 that the prediction window W slides by one each time a prediction is made. Therefore, using the input set {T−9 to T−1} defined in Figure 6, with $|W| = 7$ and $|R| = 1$, we have three rounds of predictions each using slice-1, slice-2, and slice-3. Let t denote a round, which includes generating a set of predictions using slice-1 and moving the sliding window to slice-2. Given t and $|R| = 1$, the output of the classifier function g is

$$\widehat{\mathrm{y}}_i^{(\mathrm{FBM})}(t) = g(\widehat{\mathrm{y}}_i^{(\mathrm{AFM})}(t), \widehat{\mathrm{y}}_i^{(\mathrm{IVM})}(t), \hat{\beta}_i(t-1)), \tag{5}$$

where $\hat{\beta}(t-1)$ is a bias given by

$$\hat{\beta}_i(t) = \widehat{\mathrm{y}}_i^{(\mathrm{FBM})}(t-1) - \mathrm{y}_i(t-1),$$

and $\mathrm{y}_i(t-1)$ is the observed $\mathrm{Y}_i(t)$ at time $t-1$ of user $i = 1, \ldots, N$. That is, for each round t, the bias parameter $\hat{\beta}(t)$ stores the difference between the actual label $\mathrm{Y}_i(t)[0|1]$ and estimated probability $\widehat{\mathrm{y}}_i^{(\mathrm{FBM})}(t)$, which is then passed as optional bias when the predictions for round $t+1$ ($t+|R|$ when $|R| > 1$) are made. §5 demonstrates the impact of bias parameter β on the performance of FBM.

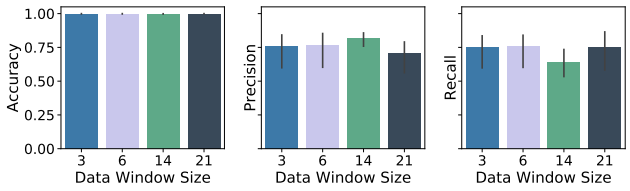
5 EXPERIMENTAL EVALUATION

We developed a prototype of PACE using open source libraries. PACE executes on a Hadoop cluster with ~45 TB memory and ~9000 virtual cores. We use Apache Pig [4] scripts to clean and join customer records from data located in both a Hadoop Distributed File System (HDFS) data lake and relational databases. The feature generators and classification models are also developed in Apache Spark [5], using XGBoost [23] ensemble libraries.

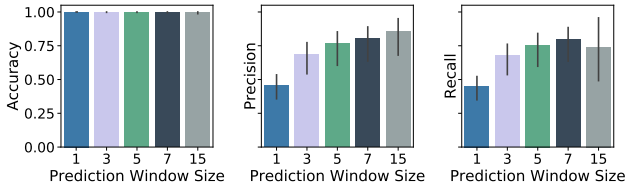
Evaluation methodology. We conducted trace-driven emulation based on historical data to evaluate the performance of PACE. We use datasets (i.e., CAI, CL, and NL) from a large cellular service provider that include two independent sample sets of data collected from June to December 2019. As shown in Figure 5, both sets are created by identifying all UIDs that have contacted care during the test period, and by randomly sampling UIDs from the NL dataset. Both sets are highly imbalanced since they contain a higher proportion of UIDs that have not called care during the observation period. The first set (collected between June and September), containing over 500K customer devices (UIDs), is used as the training set. The training set is balanced by selecting equal an number of samples from both classification classes, i.e., we select all the UIDs that have called care on a given day and then randomly select an equal number of UIDs that did not call care on this day to create the training slice for a given day. The slices are then concatenated as shown in Figure 6 to create the training set. The second set (collected between September and December) contains over 800K UIDs and is used as the test set. The test set is also transformed into slices which are then concatenated as shown in Figure 6. However, unlike the training set, the test set is not balanced by selecting equal numbers of UIDs from both classes. Therefore, the test set is highly imbalanced and contains a larger proportion of UIDs that have not called care, which closely resembles the actual care call distribution (§3.1). We use the standard statistical metrics of accuracy, precision, and recall, computed using True Positives (TPs), True Negatives (TNs), False Positives (FPs) and False Negatives (FNs). We also evaluate our models in terms of the Area under receiver operating characteristic curve (AUROC). AUROC is a more robust measure of classification performance as it integrates the prediction accuracy with all possible thresholds.

Justification of design choices. *First*, as discussed in §4.3, our models use data for the last $|W|$ time units to make predictions for the next $|R|$ time units. Therefore, the values of $|W|$ and $|R|$ should be carefully selected to balance data processing overhead, desired accuracy, and cellular provider requirements. We evaluated the impact of $|W|$ and $|R|$ on the performance of all three models (AFM, IVM, and FBM). For brevity, we omit the results of the AFM and IVM and only analyze the results of the FBM model. We found that using days as the time unit works well, and we use a time unit of days for evaluation in this paper. Figure 8a shows that $|W|$=6 yields high accuracy and precision. While higher values of $|W|$ increase precision, the processing overhead outweighs the performance benefits. Figure 8b shows that $|R|$=7 outperforms other values. We therefore use $|W|$=6 and $|R|$=7 in this paper.

Second, we compare Decision Trees (DTs), Boosted Decision Trees (XGBoost [6]), and Random Forests (RFs). We note that while Neural Networks (NNs) resulted in up to 5% performance gains in our evaluation, their results are not easy to interpret. We therefore do not use NNs in our evaluation and omit their results in the rest of this paper. The work by Diaz-Aviles *et al.* [9] makes the



(a) Impact of $|W|$ on FBM results ($|R| = 5$)



(b) Impact of $|R|$ on FBM results ($|W| = 6$)

Figure 8: Impact of $|W|$ and $|R|$ on FBM model classification performance.

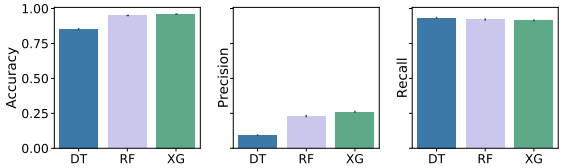


Figure 9: Comparison of DT, RF, and XGBoost models using the identity feature generator function ($\phi(a) = a$). $|W| = 6$ days, $|R| = 7$ days.

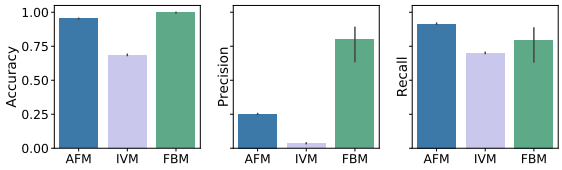
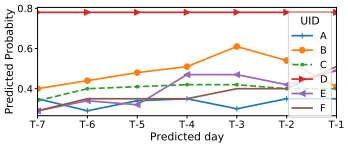


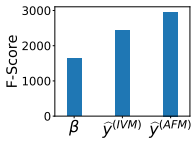
Figure 10: Performance of three models. Predictions made for 10 consecutive days using test dataset.

same choice we make for the same reason. Figure 9 presents the results using the identity feature generator function $\phi(a) = a$. We observe that XGBoost outperforms both DT and RF in accuracy and precision. DT and RF yield low precision, owing to the challenges discussed in §3. Therefore, we choose to use XGBoost. We also evaluated the performance on an unsupervised clustering algorithm to understand the efficacy of unsupervised learning in predicting user behavior, but in our analysis, we did not find any significant correlations between identified clusters and user care call behavior.

Performance of prediction models. Figure 10 shows the accuracy, precision, and recall of AFM, IVM, and FBM on 10 consecutive days. The AFM model has good accuracy and recall, but not good



(a) FBM results without bias β for six UIDs for 7 consecutive days



(b) FBM feature importance scores

Figure 11: FBM results.

Table 5: Coverage of different sampling ratios for NL.

VCR Failures	DCR Failures	VCR Success	Coverage
~0.4%	~.012%	~.01%	~16%
~0.4%	~.012%	~.04%	~18%
~0.4%	~.024%	~.01%	~16%
~1.2%	~.012%	~.01%	~18%
~1.2%	~.024%	~.1%	~23%
~4%	~.012%	~.01%	~23%

precision. IVM underperforms on all three metrics, compared to AFM. FBM outperforms the other two models in all three metrics. As discussed earlier, a key motivation for the design of FBM is to reduce the number of false positives. Figure 10 shows that the FBM model attains a 3X improvement in the precision metric, which is a measure of low false positives in the classification. We also compare all three models using the AUROC metric. Again, FBM (AUROC 0.99) outperforms AFM (AUROC 0.95) and IVM (AUROC 0.77). We note that, notwithstanding the low precision, AFM has high accuracy and AUROC, due to the high imbalance of negative class labels in the input sets.

We now take a closer look at the impact of the optional bias parameter β on FBM performance. Figure 11a presents FBM results *without* the optional bias parameter β for a set of 6 UIDs (denoted as "A", $\cdots$, "E") for 7 consecutive days (T−7 through T−1). Predictions generated by FBM have low variance without the bias parameter β, which implies that input feature vectors (**X**) and (**Z**) cannot be clearly demarcated into two categories (True/False). Updating β in Equation 5 enables the model to learn individual customer care contact behavior. During the first round of execution, the bias parameter β is not available as there are no stored predictions from previous rounds, and therefore FBM generates an unusable prediction which is ignored. Figure 11b shows the feature importance score for FBM. The FBM model learns from the results of the AFM, IVM, and the bias parameter β. We use the FBM model for our field deployment and trial experiments presented in §6.

PACE runtime scalability. PACE needs to process and track each UID experiencing a failure event during time window $|W|$, which requires processing hundreds of billions of data records per day (§3.1). This is infeasible even with a distributed PACE deployment. To avoid the overhead of tracking a massive volume of UIDs, PACE creates a list of "Candidate UID" by (under)sampling NLs within a specific analysis window. Recall from Figure 2 that nearly 75% of customers call care within 21 days of experiencing a failure.

Therefore, PACE needs to sample 21 days of NLs to ensure the candidate UID list contains at least 75% of customers that are likely to call care within next $|R|$ time units. Additionally, as noted in §3.2, customers who make a large number of voice calls are more likely to call care. Thus, PACE creates a candidate list by (a) sampling voice and data failure events in NL data, (b) sampling successful voice call records, and (c) using CL data for customers who have contacted care due to technical issues in the past 60 days. Table 5 shows the percentage of VCR/DCR records that PACE needs to process to achieve a given coverage, where the coverage (for a given $|R|$) is defined as:

$$\text{Coverage} = \frac{\text{UIDs with technical contact(s) in the candidate set}}{\text{UIDs with technical contact(s) (ground truth)}}.$$

Although the percentages of UIDs sampled from each source are small, PACE needs several million UIDs to achieve a coverage of ~16% to ~23%. We expect that including fine-grained event data such as radio-level events can increase the coverage of UIDs, thus increasing performance. However, including analysis of fine-grained data in our current evaluation can lead to significant performance overhead (§8) and therefore we do not use it in this paper.

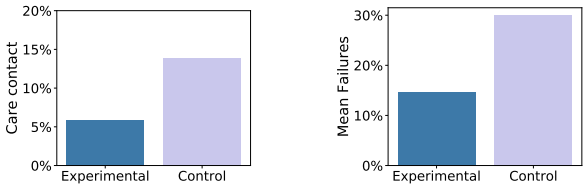
PACE is fully automated and performs the entire workflow, from data analysis to triggering resolution actions, without human intervention. Assuming that PACE needs to analyze on average ~2 million UIDs per day, the end-to-end time taken (analysis window size 21 days, $|W| = 6$ days, and $|R| = 7$ days) is a combination of time taken in (a) candidate set generation (~20 mins), (b) data preprocessing (~4 hours), (c) model scoring and predictions (~45 mins), and (d) resolution action generation (~2.8ms per UID). The entire process consumes 76.1M GB-seconds and 2,200k vcore-seconds of computing resources.

6 FIELD TRIAL IN PRODUCTION NETWORK

We deployed PACE in a large cellular service provider network in the US. We aim to understand the overall efficacy of PACE and answer the following questions: (1) How effective are the predictions and actions generated by PACE in resolving customer issues? (2) Did the actions triggered by PACE reduce the number of failures experienced by cellular customers? and (3) Is it feasible for a large cellular provider to deploy an automated framework to make online predictions and trigger resolution actions?

Field trial setup. We launched a controlled field trial in which a single instance of PACE was integrated into the production system of the cellular service provider. During this trial, PACE processes the NLs data of over 100 million user mobile devices according to the procedure described for the online phase in Figure 5 (§4). Using $|W| = 6$ days and $|R| = 25$ days[1], we randomly sample about 800,000 UIDs to be part of our experiment. These users were selected according to historical CL and NL data in the previous 28-day period. We use PACE to predict a short list of UIDs of customers who are likely to contact care due to service issues within the next 25 days. We then randomly split these selected UIDs into two groups: *control* group and *experimental* group. Since our goal here is to understand the efficacy of resolution actions generated by PACE, we only trigger resolution actions on UIDs in the experimental group and no resolution actions are triggered on UIDs in the control group.

[1]We use a large window size $|R| = 25$ days in order to cover most of 80% of customers who actually contact care as shown in Figure 2.



(a) Customers who contacted care.

(b) Mean voice/data failures per UID.

Figure 12: Results for the experimental group and control group in the field trial.

Note that PACE determines the actions to take based on domain knowledge and operational experiences provided by the cellular service provider. Determining the best next actions is beyond the focus of this paper.

Additional safeguards were also added to the production system to ensure that the mobile devices in our experimental group are exclusive from other operations and care interactions for a period of $|R|$ days.

Experimental results. Figure 12 presents the results of the field trial. Figure 12a shows the percentage of customers in experimental and control groups who contacted care during the evaluation period. We observe that: (1) Nearly 15% of customers predicted to contact care by PACE, indeed contacted care within 25 days after they experienced service issues, and (2) Resolution actions taken by PACE effectively reduced care contacts by ~60% for customers in the experimental group, compared to customers in the control group. In order to understand the efficacy of the actions in improving customer experience, we also compare the number of failures (data/voice) experienced by customers in the experimental and control groups. Figure 12b shows that resolution actions triggered by PACE effectively reduced the number of failures by ~ 51% in the next 3 days after the action was taken.

During the field trial, we observed that PACE needs to process up to 1.20 billion data records per hour during peak time. PACE consumed 6.3M GB-seconds and 297k vcore-seconds of resources and took under 45 minutes to complete.

Analytical results. Since our field trial only involves a small percentage of the entire customer base of the cellular service provider due to administrative polices to safeguard the production networks, we present a theoretical analysis of our controlled field trial results. Specifically, we tested the hypothesis that the distributions of the control and experimental groups were equal across the following two observed metrics for a period of 25 days after the resolution actions were taken: (1) the number of care contacts due to service issues, (2) the number of voice failures and data connectivity failures. Formally, given two populations $\mathcal{P}_1$ and $\mathcal{P}_2$ where $|\mathcal{P}_1| = m$ and $|\mathcal{P}_2| = n$, we validate the null hypothesis $H_0 : \mathcal{P}_1 = \mathcal{P}_2$, and the alternate hypothesis, $H_1 : \mathcal{P}_1 \neq \mathcal{P}_2$, where $\mathcal{P}_1$ and $\mathcal{P}_2$ represent the observations from the experimental and control distributions, respectively. Due to the independent samples and non-normal distribution of the data, we use the Mann–Whitney U test [14] with p-value threshold $\alpha = 0.05$, to validate the null hypothesis.

We find that for both cases (a) the number of care contacts due to service issues (U=21987.5, p-value=0.048), and (b) the number of voice failures and data connectivity failures (U=22545.5, p-value=0.028), the p-value is lower than α. Therefore we reject the null hypothesis H_0 for both cases. That is, we conclude that the $\mathcal{P}_1 \neq \mathcal{P}_2$ for both (1) the number of care contacts due to service issues, and (2) the number of voice/data connectivity issues experienced by them.

Summary. The observed reduction in the number of failures along with the reduction in the number care contacts are promising, yet these results are not exhaustive and more evaluation is required to better understand the efficacy of our predictions models. While it is infeasible to cover the entire customer base during our field trial, PACE has achieved significant improvements for individual customers, and done so without having the customer initiate a trouble report with care. In addition, PACE enables operators to move from a reactive to a proactive strategy when addressing non-outage related individual customer issues and to prioritize resolution actions based on impact of service issues. The field trial demonstrated that PACE is an effective solution to reduce the operational cost for cellular carriers.

Our field experiments also highlight some of the challenges that operators will face in deploying proactive care solutions: (a) Since only a small percentage of customers who experience failures call care to report their problems, it is important for the cellular network providers to develop algorithms to select the candidate UIDs on which predictions can be made so that network providers can prioritize their repair tasks accordingly to reduce the care calls related to these individual customer issues in addition to improve customer perceived service performance. As shown in Table 5, merely sampling the network failure and usage logs can yield low UID coverage, which can have a significant impact on the performance of the prediction models. (b) Diversity in customer behavior (as discussed in §3.1) entails that predicting the exact date on which a given customer will call remains a challenging problem. While we believe that proactive resolution actions which lead to reduction in failures experienced by customers (Figure 12b) will reduce the need for customers to contact care, operators will also be required to incorporate additional policies to prevent repeated resolution actions on the same customer device, either by a human or automated agent. (c) Network operators must carefully consider the metrics used to evaluate the efficacy of proactive care solutions. While our observations (Figure 12b) indicate that proactive resolution actions can reduce the need for reactive customer care, analyzing the extent to which reduction in network level failures will influence customer behavior remains an open problem.

7 RELATED WORK

Although there has been extensive work on automating fault diagnosis, e.g., [11] and on mining customer care logs to classify and infer problems and events, e.g., [7, 13, 16, 20], little work has attempted to correlate network data and customer care contact data, and use that to automatically take resolution action for individual customers.

The work that comes closest to ours is the work by Diaz-Aviles *et al.* [9] who investigate an African ISP and its 2G and 3G network statistics. They use an ensemble of decision trees to process network data in near real-time and predict whether a customer will contact care. Unlike our work, they only consider data services, use geographical data and detailed application information, use simpler machine learning models, and do not proactively take resolution actions. Our analysis of care calls shows that a significant number of care calls are triggered due to voice call problems, and the techniques employed by [9] cannot be directly adopted to predict voice call problems. Additionally, while it is possible for our work to leverage fine-grained data session information and geolocation information, we believe that there are considerable performance overhead and user privacy concerns in using such fine-grained data, and we therefore choose not to use it. Instead, we exploit patterns in customer behavior, such as daily usage patterns and customer care call behavior, to proactively detect and resolve customer issues using a fully automated framework.

CableMon [10] also correlates network failures with customer trouble tickets, using an anomaly detection algorithm to find abnormal events in the network and infer the subset of customers impacted by each event. Unlike our work, the focus of CableMon is on inferring failure thresholds that indicate network outages, and not on detecting individual customer-level failures caused by technical issues.

Venkataraman *et al.* [21, 22] proposed the LOTUS framework to determine users impacted by a common root cause (such as a network outage) from user feedback. LOTUS combines several modern machine learning techniques (co-training, spatial scan statistics, word vectors and deep sequence learning) in a semi-supervised learning framework. Unlike our work, LOTUS is a *reactive* approach that enables cellular service providers to relate customers who have contacted care to any possible known issues.

Finally, Iyer *et al.* [11] use connection-level traces, collected from an operational service provider, to diagnose performance problems in radio access networks (RANs). event-based performance metrics, such as connection failures and drops, they employ classification techniques, such as decision trees, to build models that explain the problem. For volume-based performance metrics, such as radio link layer throughput, they employ regression models based on physical and MAC layer information. This work is reactive and only considers RAN information and simple models. Another, less related, direction of work considers the problem of predicting churn of wireless network customers by mining social network posts [15, 17, 19].

8 DISCUSSION

Predicting *when* a customer will contact care. In this paper, we have considered the problem of predicting individual customers who will contact care for non-outage related service issues (which can be modeled as a binary classification problem) in order to prioritize proactive action for these customers. Due to missing data and lack of well-known indicators in the NL dataset, our models do not currently predict *when* a customer is likely to contact care. Instead, we identify UIDs experiencing failures during the prediction window, and classify these UIDs into customers who are likely or unlikely to contact care. While it is possible that a small percentage of customers contact care to report issues which are not observable

from the NL dataset, predicting the care contact behavior of such customers is not possible without access to rich behavioral features.

Telemetry analysis. We do not currently use telemetry data to estimate individual user service disruptions. In our experience, telemetry data aggregated at the network function level, such as at the eNodeB level, is sensitive to scale. A single rogue device generating hundreds of abnormal events can impact an entire eNodeB's performance, and falsely report that the experience is negative for all customers attached to this eNodeB. Our focus in this work is on identifying and invoking resolution actions on individual customer devices experiencing service quality degradation.

Location data. The work by Diaz-Aviles et al. [9] observes that a higher number of care contacts originates from customers who were located in congested or poor coverage areas and hence experienced a large number of packet retransmissions. One possible avenue for future work is to leverage location information in our models. For instance, a customer who is experiencing problems at their residence or employment location is more likely to contact care than a customer who experiences connectivity problems while driving in a national park during a vacation.

Data granularity. In this paper, we do not use fine-grained event information collected directly from Evolved Packet Core (EPC) control plane network functions. While we believe that fine-grained events, such as radio signaling events collected from the eNodeB, can increase the prediction accuracy, they can incur significant processing overhead. Since the radio signaling messages are routinely exchanged between user devices and cellular networks functions, such data is typically at least an order of magnitude larger than the NL data currently used by PACE. We therefore do not use fine-grained data in our current field trails.

Input data quality. We do not explore the impact of incomplete data in predicting customer behavior. While we use simple techniques such as ignoring failures caused by known network outages, completely isolating invalid and redundant data remains a challenging problem. One avenue of future work is to explore creating NLs by correlating and merging records generated by each element involved in processing user traffic.

Feature adaptation over time. The feature extractors in PACE primarily focus on voice (circuit-switched and VoLTE) and data services used by customers. While these services may originate from and terminate to devices attached to different networks (3G/4G/5G), such variations do not have a significant impact on the feature vectors used by PACE, so we expect the features of PACE to seamlessly handle the transition of existing services to 5G networks. Additionally, as 5G networks introduce changes to the endpoint/network state machine, and as customers adopt 5G-capable devices, changes in device type features may improve prediction performance.

Special events. As can be expected, customer usage, perceived service performance, and care contact patterns considerably change during holidays and special events, as well as when new devices or operating system releases become available. For example, we have noted different patterns during manufacturer device launches, and on holidays such as Mother's Day when higher call volumes are typically seen. In our future work, we plan to incorporate the impact of these events into our prediction model.

9 CONCLUSIONS

Cellular service carriers are constantly striving to improve the customer quality of experience. In this work, we proposed and described our experience with PACE, a fully automated framework to enable carriers to shift from a reactive to a proactive customer care strategy for non-outage related individual service issues. We developed three machine learning-based models, including a novel feedback model, to predict customers who are likely to contact customer care, using a combination of customer and network data logs. Using our predictions, we prioritize proactive resolution of these individual customer service issues (non-outage related) to improve customer quality of experience and to reduce customer care contacts. We report on the experience gained from a large-scale trace-driven evaluation based on real-world data collected from a major cellular service provider in the US, as well as with field trial experiments after deploying PACE into the cellular service provider's network.

ACKNOWLEDGMENTS

The authors would like to sincerely thank Eric Bonitz, Anthony Caracciolo, Zihui Ge, Ben Grizzle, Hendrik Hofman, Chandra Thompson, and Jennifer Yates for their valuable input and support of this work, and the anonymous reviewers and shepherd for their valuable comments that helped to improve the paper. This work has been sponsored in part by NSF grants CNS-1717493, OAC-1738981, CNS-1750953, and CCF-1918483.

REFERENCES

[1] 3GPP. *TS 23.228, IP Multimedia Subsystem (IMS).* http://www.3gpp.org/DynaReport/23228.htm.

[2] 3GPP. *TS 23.401, GPRS Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access.* http://www.3gpp.org/ftp/Specs/html-info/23401.htm.

[3] 3GPP. *TS 32.298, Telecommunication management; Charging management; Charging Data Record (CDR) parameter description.* http://www.3gpp.org/DynaReport/32298.htm.

[4] Apache. *Apache Pig*, 2020. https://pig.apache.org/.

[5] Apache. *Apache spark*, 2020. https://spark.apache.org/.

[6] Chen, T., and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, ACM, pp. 785–794.

[7] Chen, Y.-C., Lee, G. M., Duffield, N., Qiu, L., and Wang, J. Event detection using customer care calls. In *Proceedings of IEEE INFOCOM* (2013), pp. 1690–1698.

[8] DeFleur, M. L., and Westie, F. R. Attitude as a Scientific Concept*. *Social Forces 42*, 1 (10 1963), 17–31.

[9] Diaz-Aviles, E., Pinelli, F., Lynch, K., Nabi, Z., Gkoufas, Y., Bouillet, E., Calabrese, F., Coughlan, E., Holland, P., and Salzwedel, J. Towards real-time customer experience prediction for telecommunication operators. In *2015 IEEE International Conference on Big Data (Big Data)* (2015).

[10] Hu, J., Zhou, Z., Yang, X., Malone, J., and Williams, J. W. Cablemon: Improving the reliability of cable broadband networks via proactive network maintenance. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (Santa Clara, CA, Feb. 2020), USENIX Association, pp. 619–632.

[11] Iyer, A. P., Li, L. E., and Stoica, I. Automating diagnosis of cellular radio access network problems. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2017), MobiCom '17, ACM, pp. 79–87.

[12] Jain, V. 3D model of attitude. *International Journal of Advanced Research in Management and Social Sciences* (03 2014), 00.

[13] Jin, Y., Duffield, N., Gerber, A., Haffner, P., Hsu, W. L., Jacobson, G., Sen, S., Venkataraman, S., and Zhang, Z. L. Making sense of customer tickets in cellular networks. In *Proceedings of IEEE INFOCOM* (2011).

[14] Mann, H. B., and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics 18*, 1 (1947), 50–60.

[15] Mozer, M. C., Wolniewicz, R., Grimes, D. B., and Kaushansky, E. J. H. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on Neural Networks* (2000).

[16] Potharaju, R., Jain, N., and Nita-Rotaru, C. Juggling the jigsaw: Towards automated problem inference from network trouble tickets. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)* (Lombard, IL, 2013), USENIX, pp. 127–141.

[17] Richter, Y., Yom-Tov, E., and Slonim, N. Predicting customer churn in mobile networks through analysis of social groups. In *Proc. of SDM* (2010).

[18] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. SIP: Session initiation protocol. RFC 3261, RFC Editor, June 2002.

[19] Rowe, M. Mining user lifecycles from online community platforms and their application to churn prediction. In *Proc. of ICDM* (2013).

[20] Tan, P. N., Blau, H., Harp, S., and Goldman, R. Textual data mining of service center call records. In *Proc. of KDD* (2000).

[21] Venkataraman, S., and Wang, J. Assessing the impact of network events with user feedback. In *Proceedings of the 2018 Workshop on Network Meets AI & ML* (New York, NY, USA, 2018), NetAI'18, ACM, pp. 74–79.

[22] Venkataraman, S., and Wang, J. Towards identifying impacted users in cellular services. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (New York, NY, USA, 2019), KDD '19, ACM, pp. 3029–3039.

[23] XGBoost. *XGBoost Documentation*, 2020. https://xgboost.readthedocs.io/en/latest/index.html/.