CommSense: Identify Social Relationship with Phone Contacts via Mining Communications

Xuan Bao, Jun Yang, Zhixian Yan, Lu Luo, Yifei Jiang, Emmanuel Munguia Tapia, Evan Welbourne
Samsung Research America - Silicon Valley, San Jose, USA
E-mail: {x.bao, j3.yang, zhixian.yan, annie.luo, yifei.jiang, e.welbourne}@samsung.com

Abstract—People around the world are more connected today than ever before. By making phone calls, sending text messages and participating in online chats, mobile users are frequently interacting with their social connections through multiple communication channels. This trend is expected to continue with the emergence of immensely popular communication apps on mobile devices. Intuitively, these interactions on users’ mobile phones can reveal valuable information regarding their social relationship with their phone contacts. Understanding such relationship can help provide new services and improve users’ mobile experience. In this paper, we explore the opportunity to deeply understand these social relationship through mining mobile communication data. By building an on-device mining framework called CommSense, we show that automatically learning and understanding such relationship can efficiently support useful applications such as categorizing mobile contacts, identifying their relative importance, and automatically managing mobile contacts with very little human interference.

I. INTRODUCTION

The industry has witnessed the phenomenal growth of social communication applications in recent years. WhatsApp [1] has attracted more than 420 million active users, especially among younger generations. WeChat [2], a popular IM application with voice messaging functions, has reported a record-high achievement of 272 million active monthly users around the globe. These applications, along with existing communication channels such as phone calls and text messages, have enabled mobile users to interact with people in their social circles more frequently than ever before.

A closer look at these communications reveals a picture of a unique real-life social network. Intuitively, the interactions through mobile devices play an important role in defining and keeping social connections between individuals. Professionals make phone calls during working hours with their business connections; friends and families often communicate in the evening to share experiences and feelings; acquaintances exchange emails during holidays to broadcast updates. Unlike the friendship on online social networks, the people we communicate with through phone interactions tend to be stronger social ties in our real life.

Deriving relationship for user’s contacts can consequently lead to competitive advantages in designing mobile systems and user interfaces. This is because social relationship between a mobile user and her contacts have a fundamental impact on how she would interact with these people in the future. Designs with such insights can minimize user effort and improve user experience.

Being aware of this opportunity, we propose a social computing method without relying on conventional social network analysis. Instead of looking into the data owned by online social network (OSN) providers, which is often difficult to obtain, our system identifies users’ social relationship by analyzing their communication interactions on mobile devices. Our intuition is that most of the mobile social interactions leave footprints on the device in the form of call logs, SMS, emails, as well as chat logs (e.g., WeChat). Therefore, analyzing how such interactions are structured can provide unique insights regarding the social connections between a mobile user and her potentially hundreds of phone contacts. Moreover, since mobile communication logs are universally available on every communication device, this method can potentially work across multiple platforms without requiring users’ registrations (and their friends’ registrations) with specific social networks.

In this paper, we introduce CommSense, an intelligent on-device framework that automatically partitions mobile social connections into meaningful categories such as Family, Friends, Colleagues and Insignificant via mining mobile communication data. The concept is illustrated in Figure 1. CommSense can enable a series of compelling technologies to optimize user experience related to contact management, smart content sharing, and configuration customization. The rest of this section elaborates on some of the use cases and presents the demo applications we have developed.

Fig. 1: CommSense: Reveal social relationship via mining mobile communication interactions

A. Use Cases

CommSense enables many novel features to create interesting applications on mobile devices. We discuss some of the example use cases in this section.

1) Contact Management. Contact lists on one’s mobile device grow over time and can eventually become messy. Organizing one’s contacts on mobile device can be painful, especially for people who may have several hundreds, even more than one thousand contacts. Manually grouping and labeling each contact, even with simple group names such as “business”, “family”, can take a significant amount of time. CommSense can help solve this problem by automatically categorizing contacts into social groups according to the identified attributes of each contact. For example, a contact whom the user frequently interacts with during working hours, and always promptly responses to could be an important co-worker or manager; a contact whom the user always meets at dinner time (such information is detectable via Bluetooth) may be a family member or a good friend. CommSense can effectively identify such patterns thus automatically categorize contacts into different groups with little user effort or involvement. Figure 2(I) shows a demo application, Smart Contact Book, built...
to showcase the automatic contact management functionality. The long list of mobile contacts is arranged as a tile-based layout. Categories are prioritized based on real-time context - for example, colleagues are shown in big top tiles at work while friends may be prioritized during weekend evenings.

Fig. 2: CommSense Demo apps in Tizen, a Linux-based mobile operating system: (1) Categorize contacts into semantic groups. (2) Share contents with individual groups.

2) Information Sharing within Social Groups. Selectively sharing information among a certain group of people is a natural communication demand. A vacation photo that is perfect to share among family members and close friends, may not be appropriate to be seen by colleagues at the work place; a work document, on the other hand, should only circulate among contacts within the same organization. CommSense can intelligently help decide the range of these sharing activities. Figure 2-(2) shows another demo application that leverages the social relationship learnt from CommSense for content sharing. The application allows the user to share messages or content within certain groups in a more convenient fashion. Moreover, based on the analysis of mobile communication, the app also identifies the best communication channels (e.g., chat app, email or Bluetooth) for the sharing operation.

3) Distributed Online Social Networks. Privacy concerns remain to be a hurdle for today’s online social networks. Users usually have no choice but to trust their social network providers to handle all their personal contents. CommSense can change this situation. On CommSense enabled mobile devices, the users’ social groups are entirely derived from their daily communication and interaction patterns. The partition of the groups is completely private to the user and remains unknown to the outside world. Because of this, attribute based encryption can be applied to protect the communications within each group’s chats, posts, feeds, etc. In fact, most of the social network functionality can be supported in such a fully distributed fashion. In other words, CommSense provides a way of forming distributed social networks by generating the local view for each user without relying on cloud servers.

II. Problem Statement

The key research problem for CommSense is to accurately identify social relationship via mining multi-modal mobile communication data (calls and messages). We define social relationship categories as shown in Figure 3. All contacts are first categorized into important contacts and insignificant contacts. The important ones include Significant Others, Family, Friends and Colleagues. Notice that Significant Others also belong to Family. Insignificant contacts include Business, Acquaintance, unsaved phone numbers, and spams. In our experiment, we ask users to manually assign these labels to the important contacts in their phone book and use these labels as the ground truth. For insignificant contacts, the user only need to assign the “insignificant” label but can choose to assign additional labels if they are willing to.

Fig. 3: Social relationship category definition in CommSense

A. Data Set Overview

For data collection, we deployed an Android application (called “EasyTrack”) through Elance and oDesk crowdsourcing platforms. EasyTrack is able to collect communication logs (calls, messages) along with the associated sensing data - time, location and network. In total, we have collected data from 9899 contacts of 16 mobile users and obtained tags of contact relationship manually labeled by the users as the ground truth. Figure 4 shows the distribution of users’ home locations across multiple states within U.S. and some of them are located in Mexico.

Fig. 4: User location distribution across US and Mexico

Table I shows a detailed distribution of users’ contacts among all categories. On average, each user contributed 107 days of data. In total, 68529 phone calls and 20000 messages during recent months are analyzed. Among all the 9989 contacts, 1920 of them are labeled as important contacts (as defined before) by the user. Among the labeled contacts, there are 60 Significant Others, 638 Families, 961 Friends and 261 Colleagues. The remaining 7979 contacts are insignificant contacts (e.g., Acquaintance, Business, others).

1Some users may have labeled multiple significant others – for both cell phone and home number.
B. Challenges

Translating the idea of CommSense to reality and enabling the use case applications entail many research and implementation challenges. In this section, we briefly explain the four major challenges.

1) **Skewed distributions across categories.** As shown in Table I among the potentially hundreds of contacts of one user, usually only one contact is the significant other. A small portion of them are families and some others are friends and colleagues. In other words, the “important contact” category only accounts for a small part (20%) of the distribution, while certain sub-categories (e.g., Significant Others) only account for less than 1% of all the data. A naive classification approach may classify all the instances as negative (e.g., not significant others) and miss all the important information. A system needs to accurately pinpoint these rare but important contacts among all others to actually make a usable system.

2) **Heterogeneous behaviors.** Users behave drastically different from one another. Some heavy business users may have hundreds of contacts while many other users may only have twenty to thirty contacts; the number of phone calls can range from a few calls per month to hundreds of calls per week. These behavioral differences make users’ communication patterns towards contacts differ drastically even within the same category. For example, a business professional may call a “friend” a few times per week but a house wife may very well make tens of regular phone calls to a close friend every day. Therefore, establishing a generic model that works across all users posts a great challenge for this system.

To provide some concrete examples, Figure 5 shows the number of contacts each user has. Some of the heavy users have hundreds of contacts while others may have as few as only 15 contacts. Respectively, the patterns for these users to make phone calls and messages also show great variation. Figure 6 shows the average daily communication distribution across all users. Clearly, some users exhibit much more active communication patterns than others. On average, each user spends 17.8 minutes (with standard deviation of 19.3) in phone call and sends 18.2 messages (standard deviation 5.2), showing the heterogeneity across users.

![Fig. 5: Some users may have far more/fewer contacts.](image)

![Fig. 6: Avg. daily communication varies across users.](image)

3) **Data sparsity.** Users’ communication patterns are in general very sparse - for many contacts, often there are only a few communications across the span of several months. Even for their frequent contacts, the total duration of phone communication is also only a small portion of a user’s daily life. Therefore, a simple frequency based approach will not work directly, and sophisticated features are required to reveal the social information hidden behind the limited amount of communications.

![Fig. 7: CDF for contacts that account for more than 70%/80% of communications. Most users only frequently interact with a few contacts.](image)

Figure 7 shows the CDF of users’ interactions with their top contacts. Among all 106 users, many of them only frequently interact with a small portion of their contacts. More precisely, for most users, 20% of their contacts often accounts for more than 70% of their total phone calls. This behavior poses another challenge for CommSense since there may not be sufficient interactions between the user and some of her contacts for further classification. Moreover, phone communications in general only account for a small portion of users’ daily life. As mentioned before, on average only 17.8 minutes of calls and 18.2 messages are observed per day, scattering all over the entire 24 hours, meaning the derived spatial features need to be applicable to the sparse data.

4) **System performance.** Like any services running on mobile device, CommSense demands efficient usage of system resources. Consider users’ communication log may grow in time; it is an interesting challenge to run CommSense with an affordable resource budget.

C. Contributions

The main contribution of CommSense can be summarized into the following: 1) We present CommSense, a framework with high performance on identifying social relationship of phone contacts using heterogeneous mobile communication
data. We also provide a ranking method to further understand users’ relationship with contacts in each category. Our core method uses statistical communication patterns which are platform independent, and works regardless of what language the mobile users speak; 2) In CommSense, we design a discriminative set of interaction features that can better represent social information. These features capture high-level spatio-temporal correlations and semantics embedded in users’ daily communication. Using this feature set can significantly boost the mining performance compared with the frequency-based approach; 3) In CommSense, we propose a multi-stage classification algorithm to detect contact types incrementally. Our three-stage detection can achieve significant performance increase compared to one-step multi-class classification and can prioritize identification of important contacts; 4) We conduct extensive evaluations of CommSense using crowdsourcing data collected from more than 100 users over six months. Furthermore, we implement CommSense as a part of Samsung’s Tizen platform with high system performance.

III. SYSTEM DESIGN AND APPROACHES

Figure 8 illustrates the system overview of CommSense, integrated with Samsung’s Tizen platform. Users’ mobile interactions are stored in the form of sqlite3 databases including phone book, call log, message log, network, and location information. CommSense retrieves this information through a daemon process that periodically processes the data in batches. For each contact in the log, a series of discriminative features, such as spatial-temporal features, statistics and interaction patterns, are computed to characterize the communication behavior between the user and this contact. Following feature extraction, multi-stage classification is applied to identify the social relationship through supervised learning. Then, for each identified category, CommSense attempts to further understand the “closeness” of each contact (e.g., is this a close friend or a regular friend) through a regression-based ranking process. Notice that CommSense on device uses only data locally available from the user herself without relying on cloud assistance, which greatly preserves users’ privacy.

![CommSense System Implemented on Tizen platform](image)

A scheduler is used to duty-cycle CommSense in order to minimize system resource usage. Overall, CommSense is integrated into the Tizen software framework as a social relationship miner and exposes its social context APIs to support upper layer applications. The remaining sections will elaborate on design details with respect to the challenges described before.

A. Patterns and Features in Communication

Engineering discriminative features is one of the most important steps for performance. In CommSense, we have designed 61 features to represent users’ communications from different aspects. The key intuition is that the social relationship may affect the frequency, time and location of communications, revealing additional information from the sparse data. Table II explains the features aggregated into 10 groups.

<table>
<thead>
<tr>
<th>Feature Group Name</th>
<th># of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Reaction to Incoming Calls</td>
<td></td>
</tr>
<tr>
<td>2. Weekly Pattern Analysis</td>
<td></td>
</tr>
<tr>
<td>3. Working Hour/Inconvenient Time</td>
<td></td>
</tr>
<tr>
<td>4. Burst of Communications</td>
<td></td>
</tr>
<tr>
<td>5. Continuity and Gap Days</td>
<td></td>
</tr>
<tr>
<td>6. Adaptive Time Windows</td>
<td></td>
</tr>
<tr>
<td>7. Location of Communication</td>
<td></td>
</tr>
<tr>
<td>8. Statistics of Longer Calls</td>
<td></td>
</tr>
<tr>
<td>9. Total Duration/Frequency</td>
<td></td>
</tr>
<tr>
<td>10. Saved in Contact Book</td>
<td></td>
</tr>
</tbody>
</table>

1) Reaction to Incoming Calls: These features measure how users usually react to incoming phone calls from a particular contact. Intuitively, people are more responsive towards incoming calls from important contacts - it may not be a good idea to skip an urgent phone call from the direct supervisor while a casual call from a friend can wait. Therefore, we measure how many incoming calls from each contact are picked up or missed, how many missed calls are returned on the same day and the ratio between incoming and outgoing calls. These ratios are used to categorize users’ reactions towards calls from a contact.

2) Weekly Pattern Analysis: Certain types of communications exhibit a particular week-by-week pattern. For example, a user may regularly contact remote family members each week, even though the specific time may change due to schedule. Many conference calls or updates may also be arranged on a weekly basis. This group of features analyzes the weekly recurring pattern for communications, capturing communications per week and its variations throughout the data collection period.

3) Working Hour/Inconvenient Times: We define working hours as 9 to 5 on weekdays and inconvenient times as early morning and late night hours. Regular communications during these time windows reveal information about whether a particular relationship is profession related (e.g., colleagues or co-workers) or not. Usually, co-workers may communicate more during working hours and less so during those inconvenient times.

4) Burst of Communications: Sometimes, users may call a contact several times in a row, for example, to coordinate going somewhere or meeting at some place. Similarly, users may message each other back and forth during a lengthy conversation. We define this phenomenon as a “burst” of communications. In this feature group, we try to characterize the burst phenomenon for each contact. Currently we define calls that are within 20 minutes after the finishing time of the previous call as belong to the same burst. The features capture the total number of bursts as well as the typical burst length.

5) Continuity/Gap in Daily Communication: For certain contacts such as significant others, we make phone calls almost every day. Such continuity in communication may provide a hint about the importance of a particular contact. We define a day without any communication towards a contact as a “gap day” for that contact. These features measure how many “gap days” are there for each contact and how long the longest gap is in order to characterize the continuity of communications.
6) **Adaptive Time Window:** One limitation for fixed time window is to handle communications around the boundaries. For example, if the time window boundary is set at 5PM but one user happens to communicate around that time, these communications will fall into two windows arbitrarily. To avoid this phenomenon, we divide weekdays and weekends into adaptive time windows. These windows are defined to partition each user’s communication into separate bins thus to minimize the effect of communications near the boundaries. The communication frequency during each time window is then calculated as temporal features.

7) **Location of Communications:** For some of the communications, we have access to the associated location information - whether it takes place at home, at work, or somewhere else. These features characterize the communication location distribution with a particular contact. The most dominant location among the distribution is also computed and used as a feature.

8) **Statistics of Meaningful Calls:** The entire call history sometimes contains many calls with short durations, presumably during which the users may pick up the phone, explain that they are not available and immediately hang up. Frequent occurrence of such behavior may distort the statistics of average durations. This feature group filters out these short calls and compute statistics about longer, more meaningful calls (more than 2 minutes) only.

9) **Total Duration and Counts:** Total duration and counts of calls and messages may be a robust yet rough indicator of the strength of social relationship. These features are expected to robustly distinguish obviously close contacts from insignificant ones.

10) **Saved in Contact Book:** For classifying significant contacts vs. insignificant contacts, we also use the feature “saved in contact book”. This feature is set to 1 for saved contacts and -1 for the unsaved ones.

Together, all these features reveal the intrinsic patterns embedded in the sparse data. In general, we find that these rich features can significantly boost the performance compared to naive frequency based approaches.

**B. User-Adaptive Feature Normalization**

After computation of the rich features, normalization is adopted to account for heterogeneity in user behavior. For example, suppose we naively train a model based on heavy users that always make hundreds of phone calls per day, the resulted model may conclude that only contacts with heavy daily communications are friends. This model may not work for typical users who may only receive a few friends’ calls everyday. If we simply include both heavy and light users in a training set, the discrepancy between their behaviors may confuse the classifier.

Our intuition is that the features eventually should reflect the relative importance of a contact within the context of a given user. In other words, the model needs to decide how much a contact weighs in relation to the communications with all the contacts of a given user. Therefore, for each feature dimension in the feature vector (except “dominant location” and “saved in contact book”), all values are normalized by the maximum absolute value observed in the same dimension of the same user. This is different from traditional normalization techniques where values are normalized with the maximum observed across the entire training set. In this way, the relative ranking of each contact within each user’s record along all dimensions are preserved.

**C. Multi-Stage Classification Procedures**

Using discriminative features and per-user normalization help alleviate the problems of data sparsity and behavior heterogeneity. However, the skewed distribution among classes still remains to be a hurdle. Among over 9,000 contacts, there are only 60 Significant Others and several hundred Family and Friends, accounting for only a small portion of the population. Therefore, directly using a multi-class classifier may not perform well and may miss some of the important but rare data points. Naively adding weight will result in many false positives.

As shown in Figure 9 CommSense employs a three-stage classification approach with four decision tree classifiers and a rule-based module. The philosophy is to guarantee the performance of the most important classes first and assign the unavoidable errors to the non-essential classes. Also, this design allows the potential to accommodate multiple labels for the same contact – for example, a contact could be both a Significant Other and a Family member. In the beginning, the first step in the pipeline is to distinguish important contacts vs. insignificant contacts. This step filters out the massive number of business, marketing and other insignificant contacts. Then, during the second stage, within the important contacts, CommSense identifies Significant Others and Family members versus Friends and Colleagues. We use two binary classifiers in parallel in this step to minimize the propagation error for these two classes. For the remaining important contacts, the system attempts to separate friends from colleagues using the fourth classifier. In the end, the system makes estimates for insignificant contacts using rule-based logic learned from user studies. Details regarding the rules will be discussed in the evaluation section.

**D. Contact Closeness Ranking**

Besides categorizing contacts, CommSense also attempts to rank contacts according to their “closeness” score, measuring the strength of each social connection, with the user. Since it’s difficult for users to manually give a closeness score to every contact, we ask them to give scores to each category. Then, we design a regression-based ranking algorithm to identify the appropriate weights for each feature dimension in order to track the per-category closeness score. In the end, since contacts often have unique feature vectors, the learned parameters together can be applied to derive the closeness for each individual contact. The feature can help applications to further understand the social connections among each category. For example, among all the friends, now the closeness scores can vary due to the difference in the strength of connections, thus distinguishing close friends from regular ones.

**IV. EVALUATION AND INSIGHTS**

In this section, we provide evaluation results regarding each component in CommSense. Considering the skewness of the distribution, we define three metrics - *accuracy*, *precision* and *recall* to more accurately quantify our performance for each category of relationship $C = \{C_1, \ldots, C_n\}$. These metrics are defined as:

$$\text{Accuracy} = \frac{|\{\text{Correctly Classified Samples}\}|}{|\{\text{All Samples}\}|}$$

$$\text{Precision} = \frac{|\{\text{Contacts Belong to } C_i \cap \text{Contacts Classified as } C_i\}|}{|\{\text{Contacts Classified as } C_i\}|}$$

$$\text{Recall} = \frac{|\{\text{Contacts Belong to } C_i \cap \text{Contacts Classified as } C_i\}|}{|\{\text{Contacts Belong to } C_i\}|}$$

These metrics provide a comprehensive view of the system's performance across all categories.
For performance comparison, we derive the $F$-measure score as the harmonic mean of precision and recall: $F_{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$. We also measure the error propagation property from the confusion matrix, the mining performance during cold start, and the system resource consumption for a full evaluation.

### A. Social Relationship Identification

Figure 10 shows the performance for classifiers in identifying relationship for each category. Notice here each classifier is evaluated against its inputs. For example, the friends/colleague classifier takes non-family members from previous steps as input. Therefore, the performance for this classifier is measured for its capability of distinguishing between friends and colleagues among non-family members.

Overall, CommSense achieves over 80% precision among all categories and near 90% recall for important contacts, significant others, and friends. The recall is relatively lower for families, close to 70%. Considering the extremely skewed distribution (60 significant others and 640 families among 9899 contacts), it is challenging to identify the few families among all the contacts.

![Fig. 10: Performance per category (friends and colleagues are split by the same classifier).](image)

### B. Comparison with Alternative Approaches

We compare our performance against two alternative approaches. One approach is to use a single-step multi-class classifier instead of the multi-stage classifier. In our experiment, while keeping the feature set the same, the single-step classifier only achieves a precision of 47.64% and 68.28% for family and friends respectively, with recall lower than 50%. It performs poorly for other important categories (e.g., significant others, colleagues), showing that the skewed distribution makes this problem unsuitable for the single-step approach.

Another alternative is the frequency-based approach, where only the communication frequencies are used as features and the classification pipeline is kept the same. Figure 11 illustrates this comparison in $F$-measure. Clearly, CommSense achieves much higher performance for important categories of significant others and families while exhibits comparable performance for identifying important contacts and distinguish friends versus colleagues. This shows the effectiveness of our rich feature set for distinguishing various social relationship with phone contacts.

![Fig. 11: Comparison with frequency-based approach.](image)

### C. Effectiveness of Features

Table III illustrates the effectiveness of our feature set in distinguishing different contact categories. It lists the top five features that contribute the most at each step. Notice that, after pruning, the friend/colleague classification uses only three features. Through all these features, we can also learn interesting insights regarding how people may perceive their social relationship and behave accordingly.

Interestingly, the total communication frequency is in fact not the best feature for any category. Certain features such as ratios between weekday and weekend phone calls help capture the differences between professional and family relationship. Other temporal features like communications during certain periods also help boost the accuracy. These factors together make our method more effective than the frequency-based approach.

<table>
<thead>
<tr>
<th>TABLE III: List of Effective Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Important Contacts</strong></td>
</tr>
<tr>
<td>Total gap day - days without any communication</td>
</tr>
<tr>
<td>Ratio between incoming and outgoing calls</td>
</tr>
<tr>
<td>Total communication frequency</td>
</tr>
<tr>
<td>Median of call durations</td>
</tr>
<tr>
<td>Weekday message between 6PM and midnight</td>
</tr>
<tr>
<td><strong>Family</strong></td>
</tr>
<tr>
<td>Total call duration on Weekends</td>
</tr>
<tr>
<td>Bursts of calls</td>
</tr>
<tr>
<td>Ratio between weekend phone call duration and total call duration</td>
</tr>
<tr>
<td>Median of call durations</td>
</tr>
</tbody>
</table>
D. Confusion Matrix

Besides evaluating the performance of individual classifiers at each stage, it is also important to analyze the overall final performance. This is because classification errors in earlier stages (e.g., identifying significant others) may be carried over to later stages (e.g., identifying friends), causing errors in even high-performance classifiers. Table IV shows the final confusion matrix for CommSense, including all the propagated errors. The rows denote the ground truth while the columns show the predictions. For example, the value of family row cross friend column indicates the number of family contacts that are falsely identified as friends.

The numbers along the diagonals show that most of the relationship are correctly identified. The most errors came from the cases that identifying insignificant contacts as friends (195 among 8, 000), identifying friends as families (119) and identifying colleagues as friends (59). One reason for these errors is that the social boundaries between certain categories are not as distinct. During our user study, some users acknowledge that sometimes colleagues are also friends and some friends are treated like families. Similarly, people may rarely contact certain remote family members or friends. In fact, some of the friends or family members were called only once in several months, making it difficult to identify them.

<table>
<thead>
<tr>
<th>True/Predict</th>
<th>Insignificant</th>
<th>Spouse</th>
<th>Family</th>
<th>Friends</th>
<th>Colleague</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insignificant</td>
<td>7/701</td>
<td>3</td>
<td>75</td>
<td>195</td>
<td>5</td>
</tr>
<tr>
<td>Spouse</td>
<td>1</td>
<td>53</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Family</td>
<td>74</td>
<td>4</td>
<td>399</td>
<td>138</td>
<td>3</td>
</tr>
<tr>
<td>Friends</td>
<td>110</td>
<td>2</td>
<td>119</td>
<td>719</td>
<td>3</td>
</tr>
<tr>
<td>Colleague</td>
<td>67</td>
<td>0</td>
<td>16</td>
<td>59</td>
<td>119</td>
</tr>
</tbody>
</table>

E. Cold Start for New Adopters

To be fully functional, CommSense requires user’s communication history. In reality, users may expect CommSense to work soon after they purchase a new device. Figure 12 analyzes how CommSense performs after users start to use a device. The two curves at the top show that as quickly as one week after start using the phone, CommSense can achieve reasonably good performance of identifying important contacts as well as identifying user’s significant others. The performance for family and friends may fluctuate over the first several weeks and eventually stabilizes after 7 weeks. The fluctuation is caused by the fact that the user starts calling more and more friends/families during this period. Therefore, each week, CommSense may observe contacts that are new to the system and it is difficult to identify them right after their appearance. In the future, we expect the models learned from old devices to be saved in the cloud. Then whenever the user purchases a new device, this model can be immediately exported to bootstrap CommSense from the very beginning.

F. Closeness Ranking

After categorizing contacts, CommSense ranks contacts in each category in terms of their “closeness” to the user to find, for example, if a friend is a close friend or just a casual one. The closeness score is computed based on regression models that approximate user assigned scores using the feature set as observations. Figure 13 shows the comparison between user defined scores and CommSense scores. If the closeness score is in the range of [0, 100], the estimation error is 16.6 for in total 2341 labeled contacts collected from 106 users’ data. CommSense is able to capture the general trend of the closeness but still assigns close scores to some contacts within certain regions.

G. Rules for Categorizing Insignificant Contacts

The last component in CommSense is the rule-based logic that aims to make the best effort to tag insignificant contacts. These tags include shops/business, spam, wrong number, etc. Since the privacy sensitive information (actual phone number, contact names, etc.) are not collected, the rules are verified through crowdsourcing surveys. In the survey, the users are asked to evaluate the quality of each rule by observing their own call log – for example, how likely calls from 800 numbers are business/spam or how likely a short incoming call that happens only once is a call due to dialing a wrong number. In the end, the user report how accurate these rules are based on their observations. The results of the survey is recorded in table V based on 11 users’ feedbacks.

<table>
<thead>
<tr>
<th>Targets</th>
<th>Category</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start with 800, 801</td>
<td>Business, spam</td>
<td>70%</td>
</tr>
<tr>
<td>Incoming/short/once</td>
<td>Wrong number</td>
<td>28%</td>
</tr>
<tr>
<td>Wrong query available</td>
<td>Shops, business</td>
<td>10%</td>
</tr>
<tr>
<td>With same area code</td>
<td>Local business</td>
<td>10%</td>
</tr>
<tr>
<td>Foreign country code</td>
<td>Remote families</td>
<td>50%</td>
</tr>
<tr>
<td>Short/first called</td>
<td>Spam</td>
<td>50%</td>
</tr>
</tbody>
</table>

H. System Performance

As an on-device data mining framework, CommSense demands not only high classification accuracy but also efficient system performance. Table VI lists the average performance metrics of CommSense in terms of execution time, CPU usage, memory load, and power consumption and compare these performance metrics with other common tasks running on Samsung Tizen devices with five months of data. The average memory load of CommSense is 2.36MB and the average CPU usage is around 6.7%. Since the total execution time is around...
2.35 seconds, the overhead of CPU and memory usage is very small. The average power consumption is 227.2mW, and it consumes a negligible amount of energy considering the fast execution time. Note that in reality, CommSense only needs to be executed once per day or even once a week on one user’s data to keep the contact categorization information up to date, therefore the impact on battery life is rather insignificant.

<table>
<thead>
<tr>
<th>Comparison Tasks</th>
<th>Power Consumption</th>
<th>Memory Load</th>
<th>Execution Time</th>
<th>CPU Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommSense</td>
<td>227.2 mW</td>
<td>450 MB</td>
<td>2.35 seconds</td>
<td>97.7%</td>
</tr>
<tr>
<td>Play MP3: Radio</td>
<td>240 mW, Radio:</td>
<td>35 MB</td>
<td></td>
<td>12%</td>
</tr>
</tbody>
</table>

### V. Related Work

Analyzing the social connections between individuals has influenced research areas such as social network analytics, crowdsourcing and phone sensing [4], [5], [6], [7], [8], [9]. CommSense builds on work that falls roughly in two areas: social computing and group interaction analysis.

#### A. Social Computing

The social computing community has put a significant amount of efforts in identifying group structure in social networks [10], [11] and measuring social network evolutions [12]. The research effort can be traced back to community detection in network science and social media, where the focus is on identifying communities [13], [14]. All aforementioned social grouping approaches, however, are performed with a global view of the interaction data among all users and can apply graph related algorithms to the entire social network. CommSense adopts a user-centric approach that only uses data locally available on users’ own devices without sharing with the cloud. This helps avoid privacy breaches and does not require massive user registrations to a specific OSN.

#### B. Group Interaction Analysis

Certain interactions naturally involve a group of people – group emails, co-location, conference calls, etc. Recently, people have explored using such information for group detection. Co-location patterns obtained from Bluetooth proximity [15], [16], [17] and mobility patterns [18] can be used to infer friendships. Social media [10], caller-callee relationship [19], and emails [20] are explored to establish communication graphs among users. Instead of relying on occasional group interactions, CommSense investigates rich communication features on different modalities and works for every contact without limiting to group conversation participants.

### VI. Conclusion

Mobile communications are shaping the way people socialize with each other. Hundreds of millions active users are calling, messaging and chatting with their friends and families everyday through different channels. This paper explores the opportunity to understand a user’s social relationship via mining the fingerprints naturally left by communications. The core idea is to leverage the multi-modal information sources such as calls, messages, time, location and network and learn how these communication patterns can be translated to assess user’s social relationship towards a contact. In the end, the learned insights can be used to support services including contact management and information sharing across multiple applications. Implemented as part of the Tizen framework, CommSense shows promising results for reliably identifying important contacts with affordable system resource usage.

At this stage, CommSense certainly has many limitations. On the technical side, the current category definition is still limited and the performance also has room for improvement. We intend to explore additional opportunities beyond pattern analysis, potentially along the directions of real-time web queries and text analysis. On the social side, though CommSense is implemented completely on user’s device and shares no information with the cloud, it may still raise privacy concerns especially when users falsely perceive the results as information leakage. Though we do not have a perfect solution yet, some measures need to be taken to improve users’ perception about privacy. With these limitations, we still believe there is value in building an on-device contact mining system. With users’ contact list growing forever, it is important to keep these information well organized without relying solely on users’ effort. Functional with any mobile communication device and with no constraints in languages, We believe CommSense shows one potential direction towards solving this problem via mining communication history data.

### References


