Can Smartphones Keep Us Safe?

Pragyan Mishra
Amazon Inc.
mpragyan@amazon.com

ABSTRACT

With today's women being in the frontlines, women safety and empowerment is gaining more importance than ever.

Using Smartphones and Wearables, AI/Machine Learning technologies can ideally be used to make predictions about a person being in danger. By studying user activity patterns and combining it with location and social data, we should be able to build a safety service that is extremely useful when the user has minimal reaction time in the face of sudden danger.

AUDIENCE

This talk aims to provide takeaways for professionals of all levels.

Beginners would be able to grasp the high-level vision, understand the ecosystem that mobile device apps, underlying sensor systems and AI/ML integration can bring together to solve women issues of safety, networking, connectivity and more.

An intermediate professional would be able to gain insights into building such software – Android and iOS apps that support the vision as well as the software and hardware requirements.

Advanced audience can further evolve the talk's concept to predict what tenets of AI/ML can be modelled and perfected to learn and eventually autonomously control the various apps that address the issues of women safety and connectivity.

INTRODUCTION

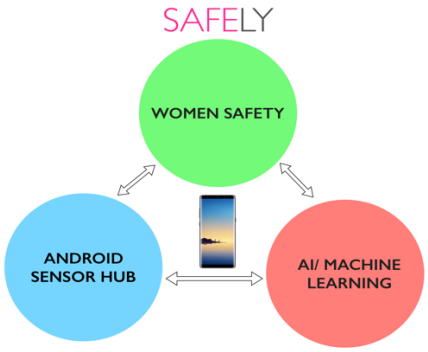
Surveys have shown that almost 45% of women are afraid to walk alone at night even for shorter distances [1]. This type of a fear of safety is not specific to women -- many men and elderly people also feel the same way. Studies have also shown that physically dangerous situations often develop suddenly and unexpectedly. While security in general has improved on University and Corporate campuses, it is still not possible to have an escort or a companion at all times. In this talk, we ask the question of whether we can use a smartphone as an intelligent safety escort to solve this problem effectively.

In today's always-connected world, we carry mobile phones and wearables such as smart watches with us at all times. These devices have grown in their ability to sense and understand the environment around them. Today, these devices have advanced sensors such as high precision accelerometers, gyroscopes, magnetometers, two-sided cameras and multiple microphones to get a multi-dimensional understanding of their environment [2]. These sensors have gotten better over the last few years in their accuracy and power consumption characteristics. Along with this, there has been a lot of recent breakthrough in areas such as Deep Learning which have resulted in improved pattern recognition on multi-dimensional and diverse kinds of data. These models also have an ability to run on today's smartphones in a battery-efficient manner.

In this talk, we explore the idea of using our own smartphone as an always-on safety escort. We have built an Android app called "Safely" as a prototype for this idea. Through this prototype, we investigate whether the confluence of technological development on smartphones sensors and Artificial Intelligence (AI) has reached a tipping point where we can build reliable, always-on smartphone-based security agents that assist the user at each and every moment of their lives.

Our prototype App, Safely, uses sensor data and cloud-based machine intelligence to detect unusual patterns and anomalies in the User's usual activity [9]. As an example,

consider the scenario of a user walking to her parking lot at the end of her workday. Instead of her normal walk pattern, however, she suddenly stops, interacts with someone and starts running. This is a pattern that Safely can reliably detect as being anomalous. Depending on the user's location (e.g. School Campus, Company Parking lot or Downtown parking) the app can proactively notify the concerned authority, law enforcement officials, nearby friends or trusted colleagues.



In this talk, we draw on our experience with Safely, to share the technical lessons and challenges in realizing a production-ready system that people can use at scale. Finally, we also discuss how the platform players such as Android and iOS can make system modifications to better support this use-case.

PLATFORM CONSIDERATIONS

The two dominant smartphone OS platforms, namely, Android and iOS both have sophisticated platform APIs to interact with and process the sensor data. Also, both platforms have the ability to run sensor processing in the background although iOS has had some limitations.

Background execution:

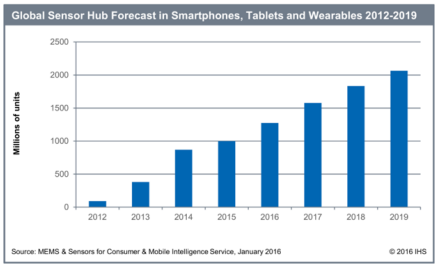
Android has extensive support for executing logic in the "background" using the concept of Android Services. The Android OS manages these services so as to balance the power consumption and functionality tradeoff. In Safely, we implement the sensor data collection and processing as an Android Service. We shall present some of the details of this architecture in the talk.

Background processing on iOS is tightly regulated. It is possible for an App to request a certain "background mode" as a user permission in order to start a background task. There are other apps that perform this kind of processing today and we certainly believe that the "reasoning" for this use-case would be strong enough to pass the Apple review for background execution.

With both Android and iOS, the App's design of background data processing has to strike the right tradeoff between functionality and power consumption. A two-phase filter, the first phase with a low false-positive rate but optimized for battery at the expense of a slightly higher false-negative rate (FIX) can be used to strike this tradeoff.

Hardware Optimizations:

Latest versions of both platforms now support what are called SensorHubs -- these are dedicated microprocessors that can be in an "always-on" state. That is, these micro-processors have a very low battery footprint that allows them to run detection algorithms all the time, especially when the main Application Processor (AP) is off. They have direct access to the sensor data and thus allow for both faster and low-power processing [5].



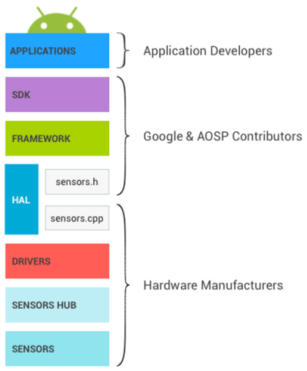


Figure 1. Layers of the Android sensor stack and their respective owners

Figure (1) [3] shows the Android architecture for a system that uses the SensorHub sub-system. The hardware manufacturers such as Samsung or Xiaomi have liberty in designing the SensorHub microprocessor which is accessed using a Hardware Abstraction Layer. This design allows for competition among OEMs to build faster and low-power sensor processors while maintaining compatibility with Apps in the Play Store.

Core Motion Framework provides reports on motion and environment-related data from the onboard hardware of iOS devices that can be used in apps [4].

Smartwatch integration:

Both Android and Apple Watch frameworks offer API to interact with the smartwatch sensors. Most watches include accelerometers, gyroscopes and a microphone. These sensors can be processed using the low-power microprocessor on the watch itself, thus allowing for faster and better detection.

DESIGN CONSIDERATIONS

In this Section, we present the architecture of our Android prototype, Safely. The insights from the design presented here would be useful for any such system that utilizes smartphone sensors and cloud intelligence for prediction purposes.

Safely, has two components: namely, the Android app and the cloud backend.

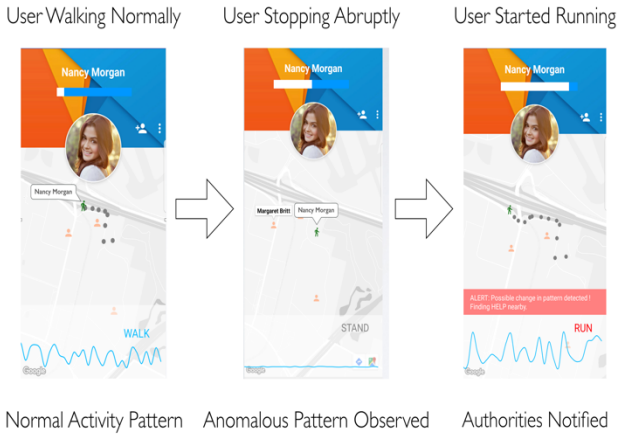
Android architecture:

The Android app consists of the SafelyService which is an Android service that collects, processes and runs detection on the sensor data. This data is collected using the Android SensorHub API [6] or in certain cases using vendor specific API such as the Samsung Smotion API [7]. The service is managed by the Android OS which allows it to be shutdown, for example, when the battery is critically low. It runs a Deep Neural Network (DNN) prediction model which is periodically trained in the cloud. We use the Tensorflow framework [8] since it has a Cloud API for training and also allows a zipped version of the DNN model to be downloaded for both iOS and Android Apps.

The second component of the app is the Safely UI which includes a Google Maps view and a graph view of the user's location and current activity. The graph view shows a user-understandable one-dimensional aggregated view of the sensor data. This UI allows the app to show the user how or why a certain prediction was made. The app also includes features for integrating a campus security alert system, for example, in order to alert a University or Corporate campus using a Text Message, a Twilio-based phone call or direct REST-API based integration.

In the talk, we shall go over an example scenario of how the detection works in Safely using real-world data.

Safely App:



Smartwatch integration:

If user has a smartwatch, this can help in two ways. First, it gives access to a supplemental set of very high-quality data, typically as the watch is worn on the wrist. The wrist also has a higher movement probability in case of a physical event. Second, the watch has a low-power sensor processor which enables fast, low-cost detection to run on the watch itself. This allows the app to make a power-efficiency tradeoff: For example, a first-pass detection algorithm can run on the watch. If this phase triggers an anomaly, the high precision (but costly) detection DNN can run on the multi-dimensional data from the watch and the phone, thus, allowing for a precise result. As long as the false positive rate for the watch detection is low, this

should result in significant battery savings without compromising performance.

The second component of Safely is the cloud backend which we shall discuss in the following Section.

Cloud-based Machine Learning Backend:

The cloud backend for Safely was built on Google Firebase and Tensorflow technologies. The Firebase provides a JSON-style database for profile and social data. It also has a key-value style CloudStore database where the training data is collected. The data is anonymized to the extent that we can collate data across users for a better-base model. The per-user personalization can then be later trained as an additional model that uses the user-specific data.

The Tensorflow component utilizes a cloud service such as Amazon Web Services (AWS) or Google Compute Engine (GCE) GPUs such as the Tesla K80s for running the training operations. These models predict anomalies in the sensor patterns which could arise from abnormal behavior. The more the data, the better would be the prediction. In order to bootstrap, this system with labeled data, we requested a set of volunteers to label the "normal" data by explicitly asking them in the app.

With regards to the DNN model, we used a standard anomaly detection model as a starting point [REF]. There are many such models available publicly to start with, while it might also be possible to use specific pre-trained models that might be available in a marketplace.

FUTURE PROSPECTS/CONCLUSION

We presented, Safely which is an Android prototype for an always-on safety agent that uses the smartphone sensors and cloud-based machine intelligence to detect any physical security threats to the user. In this process, we were able to share our experience and challenges in building such a system on today's smartphone and cloud platforms.

We believe that a useful app can be productionized on top of both iOS and Android platforms, but much more can be accomplished if the platform-owners allow for tailored provisions which are integrated deeper into the OS / hardware. This would make it faster and cheaper to run the algorithms on the smartphones.

Finally, we hope that this talk will act as a catalyst for the community to think about using our smartphones and smartwatches as an always-on safety agent.

PARTICIPATION STATEMENT

I commit to attend the conference if accepted.

REFERENCES/BIBLIOGRAPGHY

[1]http://www.stopstreetharassment.org/resources/statistics/statistics-academic-studies/
[2] https://web.stanford.edu/class/cs75n/Sensors.pdf
[3] https://source.android.com/devices/sensors/sensor-stack
[4]https://developer.apple.com/documentation/coremotion
[5] https://epsnews.com/2016/01/20/smartphones-spur-sensor-hub-growth/
[6] https://source.android.com/devices/sensors/
[7] http://developer.samsung.com/galaxy/motion
[8] https://www.tensorflow.org/
[9] https://medium.com/@curiousily/human-activity-recognition-using-lstms-on-android-tensorflow-for-hackers-part-vi-492da5adef64

BIO

Pragyan Mishra is a Software Development Engineer in Android. She works in the Visual Search Mobile Engineering Team in Palo Alto California. Her contributions include the search feature using the phone Camera, as well as Gift Card/Credit Card Scanning feature in the Amazon Android Shopping App as well as SmileCode – scan to shop that is available in Cosmo magazine's March issue.

Outside Amazon Pragyan likes to participate in tech talks, conferences and hackathons. Pragyan holds a Master's degree in Computer Engineering from University of Florida, Gainesville and B.E from BITS Pilani, India.
https://chaibytes.com/ https://github.com/chaibytes