SafeShareRide: Edge-based Attack Detection in Ridesharing Services

Liangkai Liu, Xingzhou Zhang
Wayne State University

Mu Qiao
IBM Research - Almaden

Weisong Shi
Wayne State University

Abstract

Ridesharing services, such as Uber and Didi, have enjoyed great popularity in our daily life. However, it remains a big challenge to guarantee passenger's and driver's safety. In this paper, we propose an edge-based attack detection in ridesharing services, namely *SafeShareRide*, which can detect dangerous events happening in the vehicle in near real time. The detection of *SafeShareRide* consists of three stages: speech recognition, driving behavior detection, and video capture and analysis. In our preliminary work, we implemented the three detection stages by leveraging open source algorithms and demonstrated the applicability of *SafeShareRide*. Furthermore, we identified several observations for smart phone based edge computing systems.

1 Introduction

As ridesharing services, such as Uber in US and Didi in China, have become increasingly popular, on-vehicle safety has become an important issue. There are two kinds of attacks that can happen in vehicles [1, 2, 3], drivers being attacked by passengers or passengers being attacked by drivers. To tackle this issue, Didi has applied facial recognition, itinerary sharing, SOS calling, privacy numbers, driver verification and safe driving system to provide ride safety [4, 5, 6]. However, there are still several open problems. For example, when the passenger or driver is being attacked, it may be impossible for her to press the SOS button or share the itinerary. Facial recognition and driver verification are usually only conducted for the first time when a diver registers the share ride vehicle. A safe driving system uses real-time GPS data to detect risky driving behavior. However, dangerous scenarios can still occur on vehicles with normal driving behavior. Hence, much more effort is required to ensure on-vehicle safety for ridesharing services.

In addition, the large scale of rides also poses significant technical challenges to cloud backend infrastructure. State-of-the-art safe detection systems are usually implemented on the cloud [1, 5]. However, It can be extremely difficult for a pure cloud-based approach to provide real-time safety protection services at the scale of millions of rides. Additionally, as the vehicle is moving fast, the wireless channel for applications on a vehicle may become unstable [7]. Therefore, it may take a long time to upload data to the cloud, let alone carry out the detection process. If the latency of anomaly detection is high, it is unlikely to guarantee safety in time. Therefore, a pure cloud-based safe driving system is not sufficient.

With the rise of edge computing, end devices are enabled to do more powerful computing [8, 9]. It is feasible to design and implement an attack detection platform on mobile devices using edge computing. In this paper, we propose an edge-based three-stage attack detection framework, namely *SafeShareRide*, which aims to ensure the safety of share rides. The first stage uses speech recognition to detect keywords such as "help" or a loud quarrel during a ride. The second stage is driving behavior detection. It collects driving data from On-Board Diagnostics(OBD) sensors and smart phone sensors and detects abnormal driving behavior exhibited through speed, acceleration and angular rate. The third stage is analyzing on-vehicle video recordings to determine whether there is an emergency. At the beginning of each detection period, the first two stages are running independently to capture on-vehicle danger. When attack is detected by the first two stages, video capture and analysis will be activated to process the on-vehicle video. The detection resulting from the first two stages as well as the video will be sent to the cloud or edge server. Through this three-stage detection, *SafeShareRide* can provide highly accurate detection with very low bandwidth demand of video uploading. The contributions of this paper are as follows:

- An edge-based attack detection service, *Safe-*

ShareRide, is proposed to address the safety concern in share rides. *SafeShareRide* leverages a smart phone as the edge computing platform and consists of three stages: speech recognition, driving behavior detection, and video capture and analysis.

- Our preliminary evaluation of *SafeShareRide* demonstrates the applicability of *SafeShareRide*. We provide several observations for smart phone-based edge computing systems.

The remainder of paper is organized as follows. Sections 2 and 3 depict the design and preliminary implementation of *SafeShareRide*, respectively. We present the initial evaluation of *SafeShareRide* in Section 4, and summarize the paper in the last section.

2 SafeShareRide Design

In this section, we first use an example to illustrate *SafeShareRide*, followed by a detailed discussion of each detection stage.

2.1 Example

Peter calls a ridesharing service through his smart phone. When he gets into the car, an app on his phone is launched to collect audio, driving and video data to detect abnormal events like kidnapping, fighting and quarreling. Meanwhile, some services running on the driver's phone are also automatically activated to detect abnormal scenarios for the safety of the driver. When some emergencies are detected, the related real-time video, the car information as well as the location will be sent to the cloud. In addition, a link of the video will be sent to the ridesharing service company, which in turn will take further actions, such as notifying the nearby law enforcement.

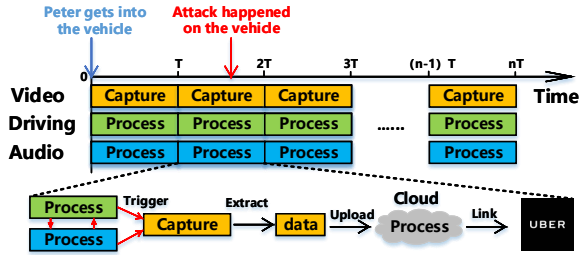

Figure 1: The time series of *SafeShareRide*.

The time series of *SafeShareRide* is illustrated in Figure 1; when Peter gets into the car, all three stages of detection are launched. For every detection period T, the audio and driving data is collected and processed to determine unsafe events. In order to reduce the demand for computation and storage resources, a trigger mechanism is implemented between these stages. If no emergency is detected during that period, the captured video data will be abandoned and no computational analysis is performed on that video. When some unsafe events are detected by speech recognition or driving behavior detection, the video capture and analyzing process will be triggered to extract the related video clips from the captured video. The compressed data will then be sent to the cloud. Video analysis will be conducted in the cloud to further examine on-vehicle safety. The video clips, along with the contextual information, such as location, time, and vehicle information, will be automatically shared with the ridesharing service provider.

2.2 Speech Recognition

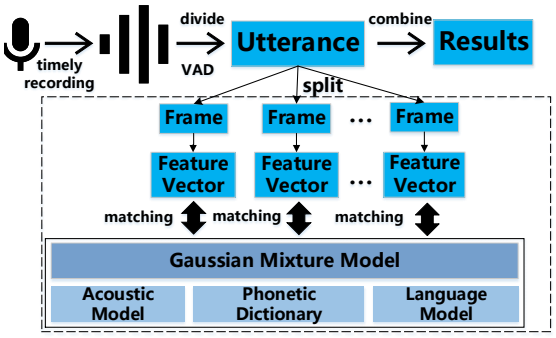

Figure 2: The speech recognition in *SafeShareRide*.

Speech recognition is the first stage in *SafeShareRide*. It is designed to detect abnormal audio, such as keywords "help" [10, 11], and abnormal high pitched sounds such as screaming and loud quarrelling. Specifically, we apply an open source speech recognition project named CMUSphinx [12] for keyword detection.

The speech recognition model that we use in *SafeShareRide* is shown in Figure 2. The model of speech is based on Hidden Markov Model(HMM) which has been widely used for speech decoding [13]. First, the microphone records the audio every T seconds(e.g., 20s) and gets a waveform. Voice Active Detection(VAD) is then used to delete the silence in the front and back of the waveform. The waveform is divided into utterances. Each utterance is split into a multitude of speech frames. Then for each frame, typically of 10 milliseconds length, a feature vector is extracted to represent the speech frame. The feature vector is used for the matching process and scream detection. Gaussian Mixture Model(GMM) is used to detect screaming and loud quarrelling [14]. Three more models are used to match the speech with words. Specifically, the acoustic model calculates the acoustic properties for each frame. The pho-

netic dictionary provides the mapping between speech frames and words. The language model restricts the word search by giving the possibility of word sequences. The matching process is to choose the best matching combination.

2.3 Driving Behavior Detection

Driving behavior detection is designed to detect dangerous driving behavior. *SafeShareRide* defines three dangerous driving behaviors: drunk driving, speeding and distracted driving [15]. They are also the three biggest causes of fatalities on the road. [1]

The data used for detection comes from the OBD adapter and sensors on the phone. *SafeShareRide* leverages the Bluetooth/WiFi communication between a smart phone and the OBD adapter to get the speed, longitude, and latitude, among others.

To detect speeding, we compare the driving speed with the road speed limits obtained from a navigation system. The detection of distracted driving and drunk driving are based on Convolutional Neural Network(CNN) [16]. In *SafeShareRide*, we use a driving dataset provided by a large automotive manufacturer. We train a CNN model on this dataset to distinguish normal driving behavior from abnormal driving behavior [17]. The trained model is then deployed on smart phones. For each detection process, the collected driving data will be divided into overlapping sliding windows according to their timestamp, such as, 0-5s, 1-6s, and 2-7s. The sliding window driving data is used as the input into the CNN model. For each sliding window, the CNN model outputs the possibility of abnormal driving behavior.

2.4 Video Capture and Analysis

Because video analysis consumes the largest computing resource, video capture and analysis is designed as the last stage of *SafeShareRide*. It will only be activated when abnormal events are detected through speech recognition or driving behavior detection. In addition, in order to reduce the latency of video analysis, only the video compression is conducted on the phone.

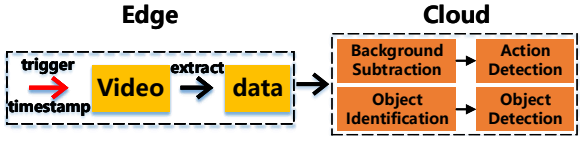

Figure 3: The video capture and analysis in *SafeShareRide*.

[1] http://www.nsc.org/Pages/nsc-on-the-road.aspx

As shown in Figure 3, video capture and analysis adopts an edge-cloud collaborative model. At the edge, based on the timestamp of the trigger signal, relevant clips will be extracted from the video and sent to the cloud. In the cloud, two kinds of detection are used for the video analysis. The first is action detection, which can detect excessive movements of the driver and passenger [18, 19]. For action detection, we first divide the uploaded video clips into many frames. Each frame needs to do background subtraction to get the outline of the human body [20]. Then we compare the outline of every pair of continuous frames to estimate the range of movement of the human body. Finally, we compare this range with the normal moving space for passengers and drivers to determine whether the movement is abnormal. Here the normal moving space can be defined as the average moving space in normal cases. The second is object detection, such as detecting dangerous objects like guns and knives [21]. CNN can be applied to recognize such objects from frames [22].

3 Preliminary Implementation

3.1 Application Framework

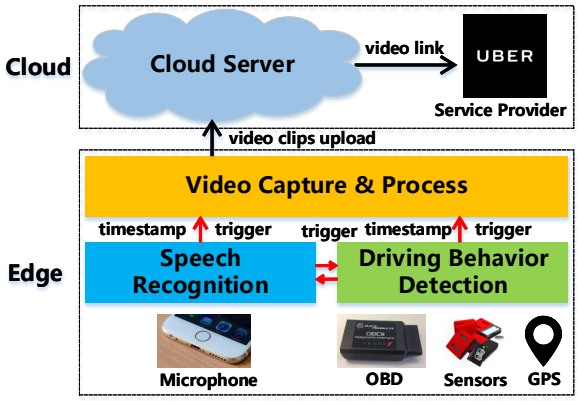

Figure 4: The framework of *SafeShareRide*.

Figure 4 shows the framework of *SafeShareRide*, which consists of two components: the first component is the edge or mobile devices, such as iPhone or iPad, and the second is the cloud.

For the edge component, a three-stage detection model is deployed to detect attacks happening on vehicle. When passengers get into the car, all three stages of detection will be activated. The detection frequency is set as 20 seconds. During each detection period, the speech recognition uses the audio information recorded by the smart phone to extract important key words. The driving safety detection utilizes the data from OBD and other sensors

to determine whether the driving behavior is normal; for example, whether the route is zigzag or the vehicle is speeding. During these two detection stages, any abnormal event will trigger the third stage detection, i.e., video capturing and analysis.

The application of *SafeShareRide* is an edge-cloud collaborative system, where the data collection and processing of audio and driving behavior data are conducted on the edge while the processing and analysis of video as well as the storage of related data are conducted in the cloud. This collaborative system is more efficient than pure edge or cloud based approaches. Because of low computing and storage requirements, the end devices can handle both the speech recognition and driving behavior detection. In addition, the end devices compress videos to save bandwidth for video uploading. The content of videos is analyzed in the cloud, as it is computationally intensive.

3.2 Detection Model

In our preliminary experiments, we leveraged open source libraries to implement all three stages of detection. Specifically, we applied CMUSphinx [12] for speech recognition, as it can work offline. We design the phonetic dictionary and language model to extract focused key words. Driving behavior detection model is based on CNN [23, 15]. The model of video capture and analysis is based on CNN, while the algorithms of video analysis are based on OpenCV [24] and Inception-v3 [21]. We set the parameters of regular moving space and video frame length.

There are several challenges in implementing the whole detection system. The first is the system overhead. We need to consider data formatting, storage as well as concurrency control, in order to reduce the overall overhead and provide real time services. The second is the detection models. We need to train the models in the cloud with large scale data sets in order to have good detection performance when deployed on smart phones. In addition, as the enforcement of GDPR [25] begins on May 25*th*, 2018, privacy protection will affect the design of the detection system to a large extent. In this paper, we have not considered the privacy issue in the system design and implementation.

4 Evaluation

In contrast to cloud-based services, edge-based services provide more stable performance and can save the bandwidth of data transmission. In order to test the performance of *SafeShareRide*, we conducted experiments to evaluate the latency, detection accuracy, bandwidth requirements, memory occupation and power consumption for both edge-based approaches and cloud-based approaches. In addition, we also conducted an energy consumption analysis when running on the phone and compared it with low energy consumption applications, such as *Gmail*.

Table 1: The experiment results of speech recognition.

Metric	CMUSphinx	Google Cloud Speech
Accuracy	73.6%	86.1%
Latency	2.279s	0.158s
Bandwidth	0KB/s	23KB/s
Energy	0.024J	0.055J

All the edge-based experiments were conducted on a mobile phone, i.e., Huawei Nexus 6P. The cloud based experiments were on the Intel Fog Node. The bandwidth consumption was measured by an app called *Trepn Profiler* [26]. The latency was calculated according to the timestamp on the phone.The detection period was set as 20 seconds.

4.1 Speech Recognition

In order to evaluate the performance of speech recognition, we defined a list of key phrases or words such as ”helping”, ”help me”, ”help us”, ”rescue me”, ”rescue us” [27]. We collected the audio data for each phrase from 10 volunteers as the test set. The CMUSphinx speech recognition and Google Cloud Speech were compared in terms of the latency, accuracy and bandwidth. Accuracy is defined as the ratio of the number of correctly recognized phrases to the total size of test set. The experiment results are shown in Table 1.

From Table 1, we can see that the Google Cloud Speech has higher accuracy and lower latency than that of CMUSphinx speech recognition. As CMUSphinx works offline, the bandwidth consumption is 0KB/s. The bandwidth of Google Cloud Speech is 23KB/s. As we don’t know how much computational resource the cloud has consumed, it may not be fair to compare the accuracy and latency. According to the results in [28], the accuracy of CMUSphinx can reach above 95% when used on a small vocabulary. The latency can be also reduced.

Observation 1: Although edge computing is very promising, it still needs optimization to become more competitive compared with the cloud based approach.

4.2 Driving Behavior Detection

For driving behavior detection, we used *TensorFlow Lite* [29] as the deep learning framework. The length of slide window was set as 5 seconds. The data collected by the OBD adapter included timestamp, longitude, latitude,

Table 2: The experiment results of driving behavior detection.

Metric	Huawei Nexus 6P	Intel Fog Node
Latency	0.264s	0.036s
Bandwidth	0KB/s	35KB/s
Energy	0.38J	0.79J

Table 3: The experiment results of video analysis.

Metric	Huawei Nexus 6P	Intel Fog Node
Latency	0.675s	0.603s
Bandwidth	13KB/s	65KB/s
Energy	0.34J	0.81J

speed, altitude, bearing, gravity, and etc. The collection frequency was 1 second. We empirically set the detection probability threshold as 0.8. If the output of CNN is larger than 0.8 will be considered as abnormal driving behavior. The experiment results are shown in Table 2.

The results show that the latency of both approaches is below one second. The edge-based approach does not have requirements on bandwidth, so it can be more stable than the cloud-based approach. When the communication signal is weak, the edge-based approach will not be affected. The energy consumption of the edge-based approach is also lower than that of the cloud-based,

Observation 2: It is more effective to train the machine learning on the cloud and deploy the trained model on edge devices.

4.3 Video Capture and Analysis

As we only uploaded video clips according to the timestamp from the first two detection stages, the required bandwidth was significantly reduced, in contrast to uploading all the video to the cloud. We set up a demo of video analysis to compare the performance of edge-based approach and the cloud based approach. In this demo, the video clips were transformed into 1280x720 (720P) and the number of frames per second was set as 30. The video data was encoded in H.264 [30] format with a baseline profile, and we configured that one intra-frame (IFrame) would be followed by fifty-nine predictive-frames (PFrames) without bi-directional frame (BFrame) because we simulated a live video stream and could not compute the differences between the current frame and the next frame. For the video analysis, we used Inception v3 to analyze the video clips.

The experiment results of video uploading and analysis are shown in Table 3. The edge-based video analysis approach is both bandwidth and energy efficient than the cloud-based approach. The latency of both approaches is close, smaller than one second.

4.4 Energy Consumption

As the battery power is limited, the energy consumption of applications is an important factor to consider when deployed on phones. We calculated the total by combining the consumption from three detection stages and measured it at different detection periods. We also measured the energy consumption of the *Gmail* on *Huawei Nexus 6P* as comparison. The results are shown in Figure 5, where *SafeShareRide*-T indicates the detection period is set as T seconds.

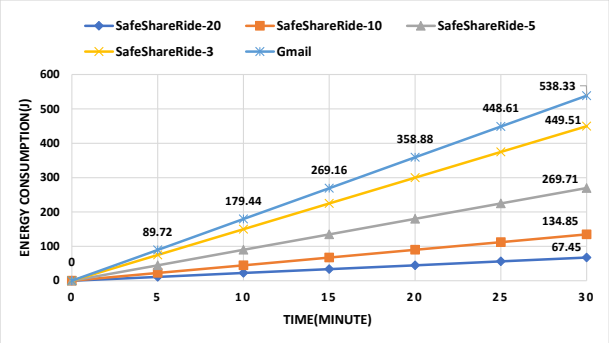

Figure 5: The energy consumption of *SafeShareRide* and *Gmail*.

From Figure 5, we can see that the energy consumption of *SafeShareRide* is lower than that of *Gmail* even when the detection period is set as 3 seconds. When the detection period is longer, the energy consumption can be further reduced.

Observation 3: Energy consumption can be further reduced by having dynamic detection periods at different detection stages.

5 Summary

SafeShareRide is an edge-based approach for attack detection. It uses three stages to provide high accuracy detection with low overhead. The three stages include speech recognition, driving behavior detection, and video capture and analysis. The trigger mechanism is used to increase the detection accuracy and decrease the bandwidth requirements of video analysis. From the three demos, each detection stage in *SafeShareRide* can perform better than the cloud-based approach. Therefore, *SafeShareRide* is an effective and efficient edge-based attack detection framework for ride sharing services. We are developing a demo of the full-fledged *SafeShareRide* service and expect it to be available soon.

References

[1] Safe rides, safer cities, 2018.

[2] Trip safety, our commitment to riders, 2018.

[3] Safety and confidence behind the wheel, our commitment to drivers, 2018.

[4] Rider and driver safety technology, 2017.

[5] Didi chuxing upgrades rider and driver safety framework, 2016.

[6] Didi upgrades driver training system with compulsory safety courses, 2016.

[7] Suk Yu Hui and Kai Hau Yeung. Challenges in the migration to 4g mobile systems. *IEEE Communications magazine*, 41(12):54–59, 2003.

[8] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016.

[9] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.

[10] Open source toolkits for speech recognition, February 2017.

[11] Christian Gaida, Patrick Lange, Rico Petrick, Patrick Proba, Ahmed Malatawy, and David Suendermann-Oeft. Comparing open-source speech recognition toolkits.

[12] CMUSphinx: open source speech recognition toolkit, 2017.

[13] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[14] Luigi Gerosa, Giuseppe Valenzise, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. Scream and gunshot detection in noisy environments. In *Signal Processing Conference, 2007 15th European*, pages 1216–1220. IEEE, 2007.

[15] Chunmei Ma, Xili Dai, Jinqi Zhu, Nianbo Liu, Huazhi Sun, and Ming Liu. Drivingsense: Dangerous driving behavior identification based on smartphone autocalibration. *Mobile Information Systems*, 2017, 2017.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] Zhongyang Chen, Jiadi Yu, Yanmin Zhu, Yingying Chen, and Minglu Li. Abnormal driving behaviors detection and identification using smartphone sensors. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pages 524–532. IEEE, 2015.

[18] Massimo Piccardi. Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE, 2004.

[19] Andrews Sobral and Antoine Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21, 2014.

[20] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1219–1225. IEEE, 2009.

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[23] Ravi Bhoraskar, Nagamanoj Vankadhara, Bhaskaran Raman, and Purushottam Kulkarni. Wolverine: Traffic and road condition estimation using smartphone sensors. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–6. IEEE, 2012.

[24] OpenCV: Open source computer vision library, 2018.

[25] General data protection regulation, 2018.

[26] Trepn Profiler6.2: Qualcomm innovation center, inc., 2016.

[27] Javier Ramirez, Juan Manuel Górriz, and José Carlos Segura. Voice activity detection. fundamentals and speech recognition system robustness. In *Robust speech recognition and understanding*. InTech, 2007.

[28] Hassan Satori, Hussein Hiyassat, Mostafa Harti, Nouredine Chenfour, et al. Investigation arabic speech recognition using cmu sphinx system. *Int. Arab J. Inf. Technol.*, 6(2):186–190, 2009.

[29] Introduction to TensorFlow Lite, 2018.

[30] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.