TITLE :

Predicting distracted drivers and sending corresponding communication signals to nearby devices.

ABSTRACT

Reckless driving plays a critical role in changing people's lives. Thus, it is critical to detect reckless drivers and inform other drivers. Several papers detect dangerous drivers efficiently. However, few papers address how to build a system to inform other drivers about the dangerous driver. In this presentation, we are going to present several recent papers that address the issue of detecting reckless drivers using machine learning algorithms and show the limitations of these existing methods.

MOTIVATION

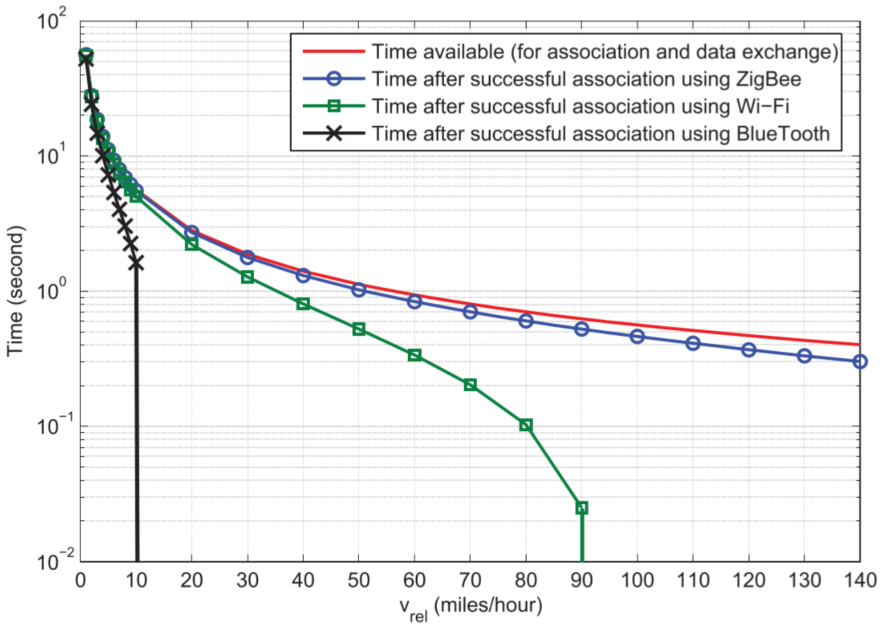
Most of the vehicular accidents take place due to careless driving and this also affects the nearby drivers. From our research, there is no current system or application that can detect and warn the nearby drivers. Also, according to one of the studies, if the drivers were warned within 1.5 seconds then it would have saved them from various accidents. The project takes into account various scenarios like detecting a collision, emergency or sudden brakes, overtaking or changing lanes without any indicator. To instantly communicate with the nearby car/devices it will be inefficient to use the cloud services to communicate between two devices. So we implemented the communication with the help of Bluetooth(It can also be done with the help of other technology like Wifi Direct, ZigBee, etc.)

RELATED WORK

Vehicle-to-Vehicle Connectivity and Communication Framework for Vehicular Ad-Hoc Networks

Paper presented about V2V communication and more specifically on the underlying parameters. It suggests two main parameters: Association Time and Data Transfer time. In association time it is the total time to establish a connection between car and the later one is to transfer the data. The total time the mobile devices perform these operations depends on the relative speed of the car and also the transmission range. If the transmission range is large then the two devices are going to be in range for a longer time.

The author compares ad-hoc networks like Bluetooth, Wifi, ZigBee and those plots that show interesting findings as shown in figure 1. In the later section, we describe our solution and architecture on how we can use the idea of multiple hops to increase range.



Invisible Sensing of Vehicle Steering with Smartphones

We also went through a few papers on using the underlying sensors in the phone to achieve our project goals on identifying the various events that need to be warned to other drivers. This paper suggested many findings and how we can use them in our application. One of the things that we were able to develop was collision detection and sudden brakes using a sensor. Paper also suggested how the lane changes or overtaking of the car can be detected using sensors and algorithms for it. The authors also compared the sensor-based system with a car and it showed that it is 11 times better than the visual-based sensor and also provides desired results in the extreme weather condition.

PROJECT BODY

PROBLEM STATEMENT :

To devise a system that can recognize signs of dangerous/impaired driving, using non-intrusive methods and detect accidents before they happen, send signals to the potential principle of the accident, and nearby drivers or pedestrians. This involves communication between smartphones and or vehicles in motion in minimum time. Also use mobile sensors to detect impair driving.

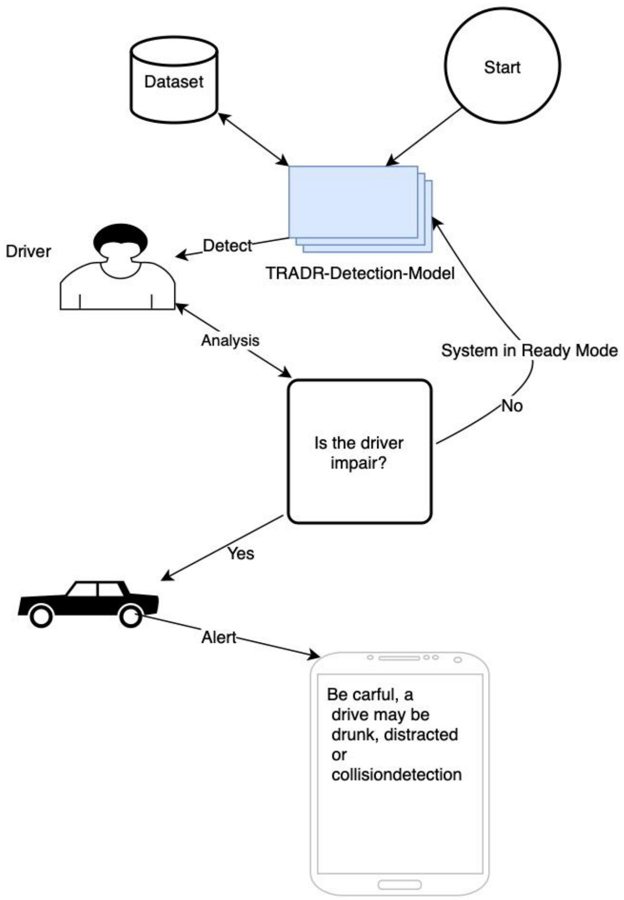
DATASET FOR MACHINE LEARNING

In this project, we got a dataset of distracted drivers with 10 classes. These images were RGB images. The 10 classes are as follows:

- c0: safe driving
- c1: texting - right
- c2: talking on the phone - right
- c3: texting - left
- c4: talking on the phone - left
- c5: operating the radio
- c6: drinking
- c7: reaching behind
- c8: hair and makeup
- c9: talking to passenger

Each class had approximately around 2200-2300 images so that the training set was balanced. The images were of 640x480 size. We had a testing data of approximately 12.5k images which were unlabeled. The major goal out of this was to predict if a driver is distracted or not.

SYSTEM OVERVIEW



DATA PREPROCESSING

As in any machine learning problem, the toughest part is the preprocessing step and we ran a number of experiments over here. Observe that our images had a size of 640x480, which means every image had 307,200 dimensions. According to the curse of dimensionality principle, the data required to train a model grows exponentially with the number of dimensions. Thus, this observation led us to first reduce the dimension of the image before training this image. From this observation, we reduced our image from 640x480 to 224x224. Once the data reduction is done, it is important to normalize our images. We know that the pixel values range from 0-255, thus when we calculate errors, the weightage of the error corresponding to a pixel of 255 would be much larger. So we get all the pixel values into the range of 0-1 by dividing our image matrix by 255. This ensures uniformity while calculating errors during backpropagation

COMMUNICATION TECHNOLOGY

The most common ways in which two devices can communicate is either via some infrastructure or with the help of ad-hoc networks. For the infrastructure based system, each car needs an access point to communicate via the internet. Here the mobile phone can be used as a gateway to communicate with the internet. If we don't consider the mobile phone then we need to configure the access point

and also provide IP addresses from the DHCP but here in order to communicate we need high bandwidth since that's where the infrastructure-based technologies are optimized especially in LAN.
On the other hand, the Bluetooth which is an ad-hoc network can automatically configure itself and thus provides flexibility. Since we need to send a message to a larger range and there is a limitation of range in Bluetooth. As per our experiments, we were successfully able to receive a message within a range of 10 meters. It also depends on various devices. In our implementation, we plan to send a message via hops such that each device will forward to other devices in its range. So this covers a broader area.

IMPLEMENTATION

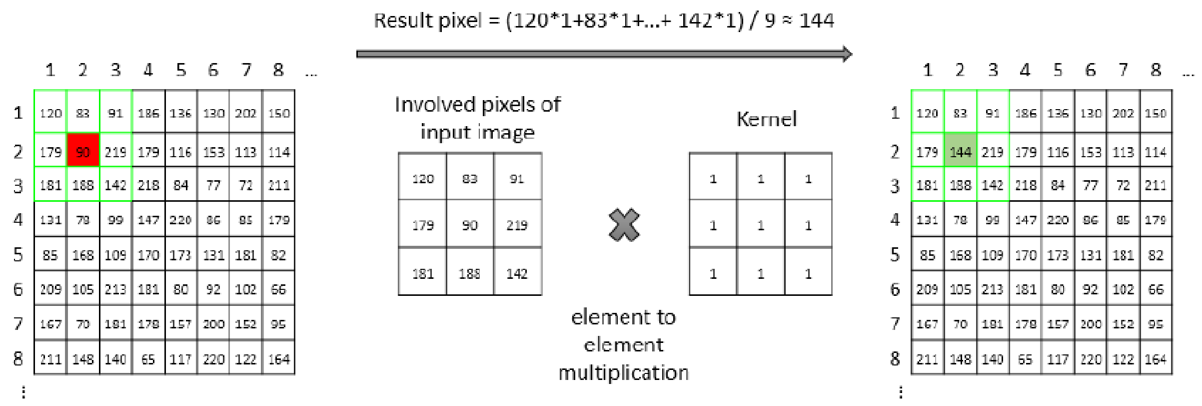
We implemented the application in Android. The following are the components of our project.

A. Communication: The following component will help in communicating with drivers via Bluetooth messaging. The message transferred between the car includes Car Model, Car Number, Color, Special Identity, latitude and Longitude and type of warning. The message is compressed and abbreviated to decrease the size of the data packet. (For Example Toyota as '*T'*).
B. Location: The message sent between the car contains Location coordinates. This can be used to calculate the distance between two cars and also show the location of the car so the driver can take some precautions. Here we are using Google maps API for showing Map and the location.
C. Sensor: With the help of sensors in mobile we implemented some components like collision detection, sudden harsh brakes, also sharp overtake. This was implemented with the help of the tools provided for android app development.

For identifying images using machine learning we built CNN in python using keras and tensorflow. Following is working and implementation of CNN.

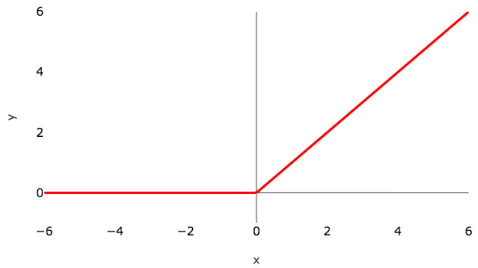
Choice of Model : The state of art as of today for any image classification problem is the Convolutional Neural Networks, these CNN's also observe the spatial relationship of each pixel. The CNN we have made has 13 layers. Below is a detailed explanation of the type of layers used:

Convolutional Layer: This is the heart of CNN. Basically, here we take an image and we convolve it with respect to its kernel. We take multiple of these kernels, for example, the number of filters used in the first layer is 64. These filters capture spatial relationships amongst the pixels of the image.



The above diagram depicts how the convolution is working, observe that the kernel size is 3x3. Thus, our entire image is divided into subregions of 3x3. Convolution is done by multiplying each individual pixel with the corresponding pixels in the kernel and adding them all up as shown in the picture. Later we divide by the number of pixels in the kernel. The number that we get will represent the entire area and thus we say that the spatial relationship would be observed.

RELU: After convolution is done, these new pixels obtained goes into the rectified linear unit. All the negative pixel values would become 0 when we pass through this function. Basically, relu is simply,



Max Pooling: After RELU, our image goes into the max pooling layer. The max pooling layer will shrink the size of the image and only represent the values corresponding to the maximum value in a particular region. The size in our max pooling layer is 2x2. This makes sure that even if the image is rotated, by a few degrees we can still capture the relationship of the pixels.

Dropout Layer: Basically, a dropout layer is used so that our model does not overfit to the training images. For example, there is a high chance of learning the driver, instead of the actions made by the driver. The dropout layer will ensure that only important features would be considered while training our model. What matters is the action of the driver and not the driver himself/herself.

Softmax Activation Function: The softmax activation function at all points would take the input and return the probabilities of an image belonging to a certain class. According to the max probability, we classify the image into one of the given classes

EVALUATION METRICS

1. Measure time to send a message from one device to another. It is calculated in the following way: When a message is sent from one device to another we also send a timestamp which is the time when the packet was created and we then take another timestamp when it is received on the other side. The difference indicates how much time it took to pass the message. The observation is shown in the following section with various scenarios.
2. To check if the data packet size affects the connection time we performed by testing with data of different lengths. (Although from our observation we did not find any significant changes with change in data size of max 100 bytes). Our application is expected to deliver all messages within 100 bytes.
3. Since there will be external forces like too many devices connected with the same device and also a different configuration in the many phones so we created a heterogeneous environment and obtained the result.

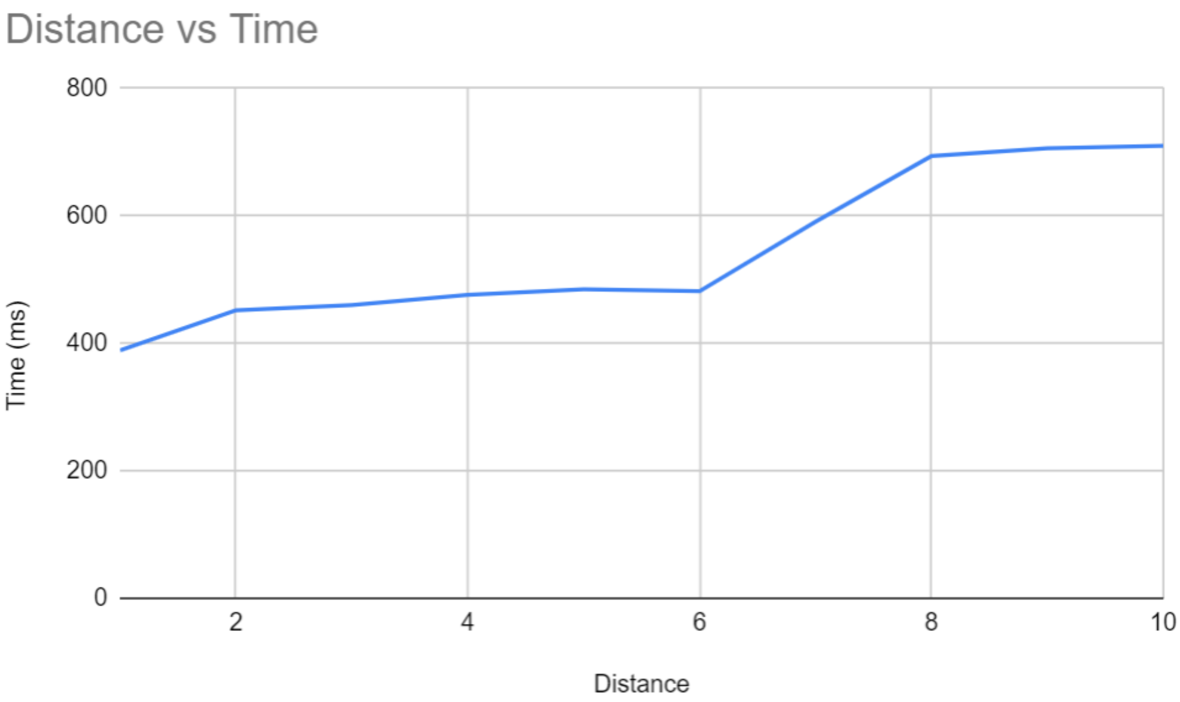
SCENARIOS

Since we were building a CNN, the amount of computation done in CNN is huge. Thus, we needed a lot of computation power. Due to this we chose to run all our experiments in google colab and mounted all our images into google drive.We used the Keras library with tensorflow as its backend. We ran 30 epochs and got an accuracy of about 98% for our validation dataset.

RESULT AND ANALYSIS

A. Distance between two devices vs time to receive message

The main testing was conducted for V2V communication via the ad-hoc network i.e. Bluetooth. Below is the plot for the distance between two devices and time is taken to receive the message at the other end. The readings are almost constant as long as it is within the range



Approach: Two android devices were placed at an 'x' distance from each other and reading in the interval of 1 meter was performed. The message was sometimes lost after 9 meters.

B. Reading from mobile sensors

Here testing of the sensor was conducted. The phone was tested on various surfaces rough and also hard ground. From the below plots, it can be seen how the accelerometer reacted for various events. For surfaces/road which is completely damaged can give false negative i.e. false warning since the vibrations will be considered as some kind of collision.

Fig -1.1 changing of lane with gyroscope

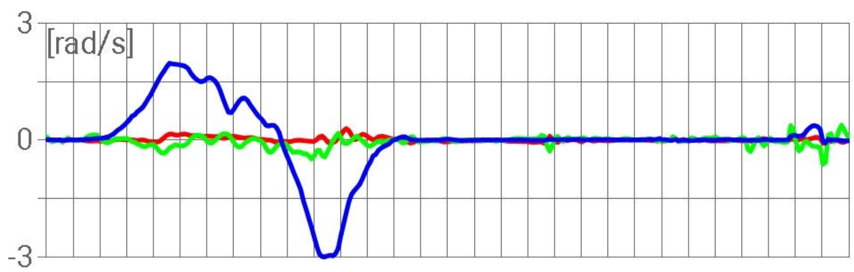


Fig - 1.2 Detecting collision using accelerometer

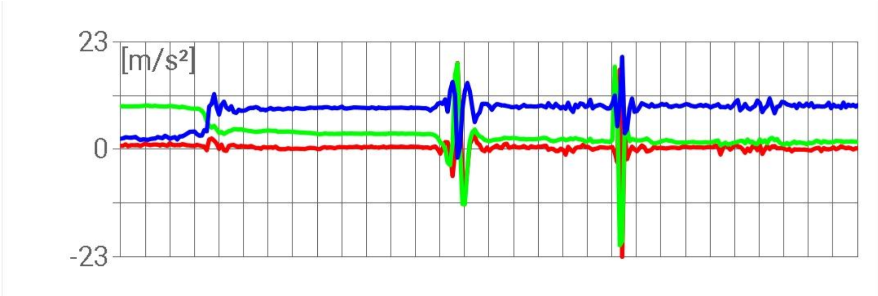
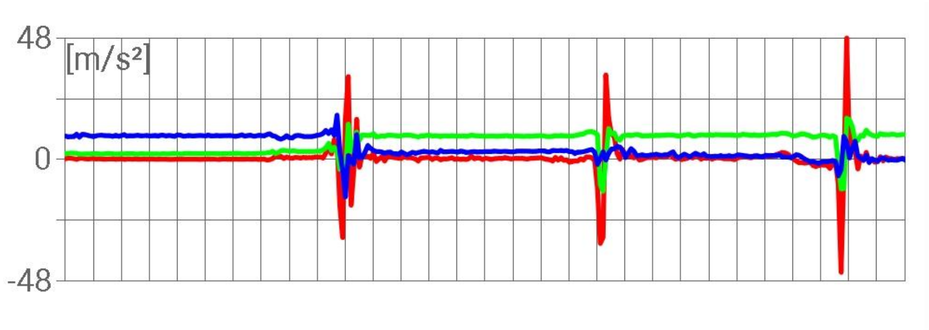


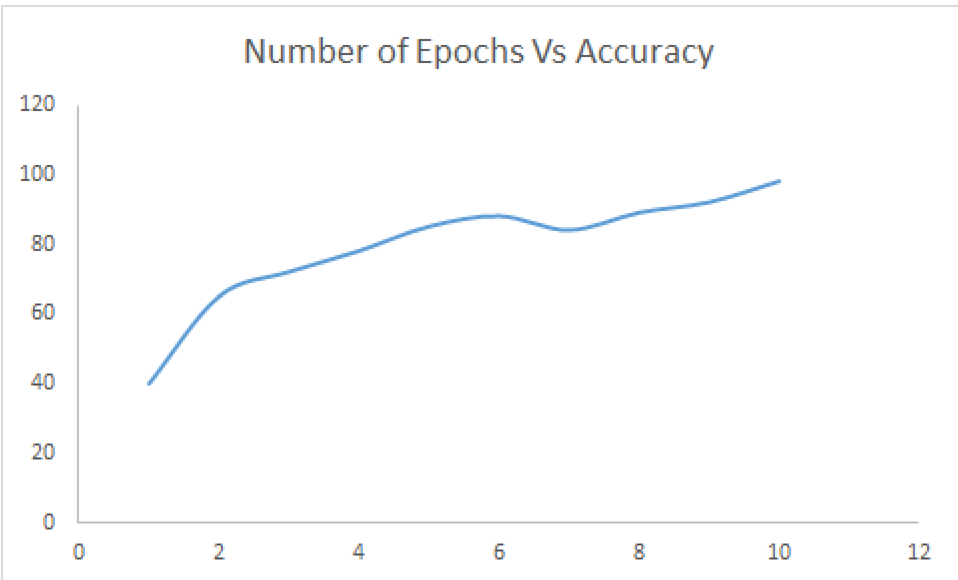
Fig - 1.3 Sudden turn



C. Phone compatibility

We performed some cases like one device being connected to any device and then sending a message. (This thing may work for only some phones). The message passing was not affected by this since the underlying implementation is more like client-server and it occupies a different channel for communication

D. Epochs Vs Accuracy



Above is a graph for the number of epochs vs accuracy and one can see that the accuracy is increasing as we increase the number of epochs. This is due to the fact that as we increase the number of epochs, the model starts to learn the weights and corresponding parameters with respect to the weights.

EXPECTED RESULT

Although the application was developed for testing and as a working prototype. The results with Wifi Direct as an underlying Ad Hoc network could have been much better. Maybe in the future, there will be some better technology on the phone which would yield a better solution.

FURTHER WORK

One of the challenges we faced while implementing was to achieve multi-hop message passing in the Android App development. Another challenge that needs to be worked in the future is to understand mobile sensors. From our observation, it gave a similar graph when similar action or event took place (For example when the car changes lanes then it will show a bump in the graph).

Also, this project aimed to create an application for mobile devices but what if a special hardware device were installed across the road (Example signals) which could transmit the message to cars. This hardware could be developed with better configuration and high range.

Since basic model is ready, we plan to add weights and classify our model predictions in 3 categories:

1. Safe
2. Moderate dangerous
3. High Danger

TASK DISTRIBUTION

Our project consisted of two major components: Detecting impaired drivers and others was using V2V communication to warn drivers. For this we divided teams as follows:

Deep, Rohan worked on the V2V communication part in Android and worked on Google Maps API and Bluetooth Message passing.

Tirth, Raed worked on detecting drivers using machine learning and neural networks. Implemented in python using keras and tensorflow

Ahmad worked on the sensors part, like working on developing algorithms to figure out the pattern.

Overall all of them contributed equally and had discussed the project together with their valuable inputs.

REFERENCES

[1] Zhang, L., Yan, L., Fang, Y., Fang, X., & Huang, X. (2019). A Machine Learning-Based Defensive Alerting System Against Reckless Driving in Vehicular Networks. IEEE Transactions on Vehicular Technology, 68(12), 12227-12238

[2] Vijayan, V., & Sherly, E. (2019). Real time detection system of driver drowsiness based on representation learning using deep neural networks. Journal of Intelligent & Fuzzy Systems, 36(3), 1977-1985.

[3] J. Smolka and M. Skublewska-Paszkowska, "A method for collision detection using mobile devices," 2016 9th International Conference on Human System Interactions (HSI), Portsmouth, 2016, pp. 126-132.

[4] Mehta, V., Katta, S. S., Yadav, D. P., & Dhall, A. (2019, October). DIF: Dataset of Perceived Intoxicated Faces for Drunk Person Identification. In 2019 International Conference on Multimodal Interaction (pp. 367-374).

[5]. R. Berri and F. Osório, "A Nonintrusive System for Detecting Drunk Drivers in Modern Vehicles," 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), Sao Paulo, 2018, pp. 73-78.

[6]. Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G. Shin. 2015. Invisible Sensing of Vehicle Steering with Smartphones. In Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys ’15). Association for Computing Machinery, New York, NY, USA, 1–13.

[7]. D. B. Rawat, B. B. Bista, G. Yan and S. Olariu, "Vehicle-to-Vehicle Connectivity and Communication Framework for Vehicular Ad-Hoc Networks," 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems, Birmingham, 2014, pp. 44-49.