

SafeDrive: An Autonomous Driver Safety Application in Aware Cities

Koosha Sadeghi^{*}, Ayan Banerjee[†], Javad Sohankar[‡], and Sandeep K.S. Gupta[§]

iMPACT Lab, CIDSE, Arizona State University

Tempe, Arizona, USA

Email: {^{*}ssadegh4, [†]abanerj3, [‡]j.sohankar, [§]sandeep.gupta}@asu.edu

Abstract—Commercially available wearables and apps that convert mobile devices into data collection hubs can be used to implement smart applications in aware cities. In this paper, we consider wearable devices on various human users as a networked cluster of computing power and information source in an Internet-of-People architecture. Applications can be developed to perform computation on this data and gain group level aggregate inferences and provide feedback. We propose “SafeDrive”, an autonomous transportation application, that estimates mental fatigue of a driver using brain sensors, predicts collision probability by fusing car parameters with driver mental state, and issues feedback just in time to avoid accidents. However, significant challenges exist with respect to ensuring safety, accurate context computation and real-time operation, and sustainability, resource efficiency. In this regard, we present the “HumaNet” framework that consists of a middleware installed in mobile devices for developing aware cities applications. HumaNet applies model-based requirements checking approach for off-line analysis and optimization of a given application. Further, it applies context-based requirements checking to decide how to use different types of computation resources to satisfy safety and sustainability requirements in run-time.

I. INTRODUCTION

Advances in the Internet of Things (IoT) domain is making city infrastructure smarter, by incorporating context awareness in different domains including transportation, health, education, and energy management. Context awareness in aware cities is obtained through collection of data from the environment including its inhabitants using sensors, inferring knowledge from the data, and using that knowledge to provide automated services [1], [2]. To this effect, deployment of Internet enabled wearable devices on human body such as the Apple watch, the Google glass, and smartphone has converted the human user to a human server, a rich repository of environmental, and physiological data [3]–[5]. The IoT concept has morphed into the Internet-of-People (IoP) that enables sharing of the knowledge through community networking of human servers for collaborative applications such as traffic condition reporting or crowd driven emergency response [6].

In this paper, we propose SafeDrive, a context aware transportation safety application, that uses community monitoring to determine mental fatigue levels of drivers and provides them feedback on impending collisions. SafeDrive assumes that each driver on the road is equipped with a brain sensor that connects to their smartphone. The smartphone is also a

hub for other car sensors such as the wheel speed, the rear view, or blind spot cameras. The smartphone sends the mental fatigue level along with car speed and neighboring car details to a central server. The server computes impending collision based on the mental fatigue levels of other drivers on the road as well as their current driving status and alerts individuals using alarms or warning messages. SafeDrive makes city transportation context aware by combining knowledge from different domains such as mental states of drivers and mechanical states of a vehicle, transferring them to different computational and decision making hubs in the IoP infrastructure, and processing them in a collaborative environment to derive community based inferences, which influence decision making. For SafeDrive to be useful, it needs to satisfy three main requirements: a) the mental state detection system has to be accurate, b) the feedback on collision has to be real-time, and c) it needs to be energy efficient on smartphone.

Previous works on driver drowsiness detection systems are based on information from an individual (i.e. driver). In [7], facial expression and photoplethysmograph (PPG) data are combined to measure driver alertness. They reach 83% and 94% true detection rates using two features (i.e. percentage of eye closure and PPG power spectrum density) and two additional features (i.e. average eye closer speed and heart rate variability), respectively. In [8], set of medical grade sensors are used to record electroencephalogram (EEG), electrooculogram (EOG), and electrocardiogram (ECG) signals to detect drowsiness with 95-97% accuracy. In [9], skin conductance and oximetry pulse are monitored to detect mental fatigue and they reach around 93% accuracy using neural network approach. In [10], a four channel wireless EEG sensor is used to detect drowsiness and an accuracy of 74.6% is obtained. At last, in [11], they use Neurosky [12] EEG sensor and support vector machine algorithm for drowsiness detection and reach 88.8% accuracy in the best case. These works suffer from at least one of these four limitations: 1) usage of medical grade sensors with tedious setup procedure (reducing usability), 2) combination of several types of sensors and features (reducing usability), 3) computationally intensive processing (hampering real-time operation and energy efficiency), and 4) low performance (potentially undermining safety). These shortcomings reduce usage feasibility in on-line mobile settings.

In SafeDrive, information from all drivers on a route is collected and processed to improve the performance. Applying

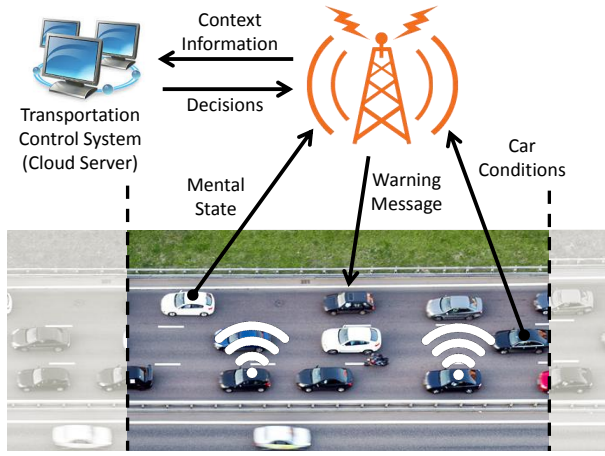


Fig. 1: Operation of SafeDrive application using HumaNet.

IoP approach in developing SafeDrive makes it feasible to: 1) exploit easy-to-use and commercially available sensors such as Neurosky or Emotiv [13] (limited number of channels and low signal-to-noise ratio of these headsets are challenges toward reaching high accuracy), 2) use just one type of signal features (i.e. variations in the frequency domain), 3) deploy a light weight mental fatigue detection algorithm (based on a Markov chain model), and finally, 4) obtain an average of 91% accuracy to detect mental fatigue (in five trials of one subject). These points make SafeDrive usable as well as accurate. It is noteworthy that evaluation on a larger pool of subjects is required for real-world usages.

To enable safety and sustainability assured IoP based implementation of SafeDrive, we propose HumaNet. It is a framework that applies *model*-based [14] and *context*-based [15] requirements checking to ensure safe and sustainable operation of aware cities apps such as SafeDrive. In model-based checking, HumaNet runs models of the app to determine its specifications (e.g. response time and power consumption) off-line. Model-based checking also indicates optimized configuration of the app to be executed with high accuracy, real-time, and energy efficient way. Off-line monitoring may not be sufficient by its own to guarantee safety and sustainability for the running app. So, context-based checking monitors execution of the app on run-time and according to the context decides whether to offload the computations on the central server to satisfy requirements. To implement SafeDrive app based on HumaNet, an application specific middleware is downloaded into each participating smartphone. In model-based checking, temporal logic is used to optimize the execution of SafeDrive. Afterward, in context-based checking, SafeDrive is monitored in run-time to check if safety and sustainability requirements are met. Otherwise, the computation is offloaded on a central cloud server.

II. SAFEDRIVE: DRIVER SAFETY APPLICATION

As shown in Figure 1, two types of information from each driver are sent to the transportation control system. The first

type is driver's mental state (i.e. alert or mentally fatigue) and the second type includes car conditions (i.e. its speed and location). The control system server processes the collected information, and based on the analysis results, an appropriate warning feedback will be sent to driver. In our system, for triggering an alarm tone, we consider two scenarios, 1) an individual mental fatigue level is lower than a threshold, and 2) a fatigue driver is nearby or probable collision is predicted.

In this sense, as seen in Figure 2, a wearable device on a human user has to provide three interfaces: a) an interface to connect to available sensors and create databases for each of them, b) a platform to execute application executables, and c) a webserver that can be accessed through the Internet and hosts the raw data or the output of the applications executed in the platform. SafeDrive app executables (Figure 2) consist of two parts: 1) client side that executes a mental fatigue detection algorithm (for the first scenario), and 2) server side that receives mental states from client side, runs collision detection algorithm, and sends back feedback messages to client side (for the second scenario). On the client side, the EEG sensors capture signals and send them to the smartphone. The collected data is processed and the alertness of the driver is determined according to the preset threshold. Recognition of fatigue state will trigger an alarm tone. Furthermore, her mental state is sent to the central control system (i.e. server side app) along with her car conditions. In the server side, the transportation control system sends a warning message to a driver in case of the presence of a fatigue driver nearby or prediction of a probable collision in her vicinity. In sections IV-A1 and IV-B, we describe the details of measuring mental fatigue level and detecting probable collision, respectively. To measure the mental fatigue level of a driver, we need to train our algorithm to determine a threshold before using the app during driving

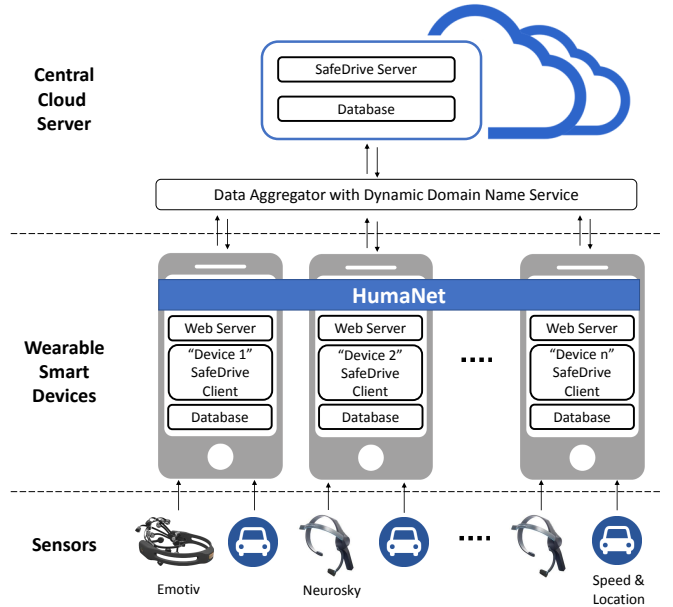


Fig. 2: The autonomous driver safety system model.

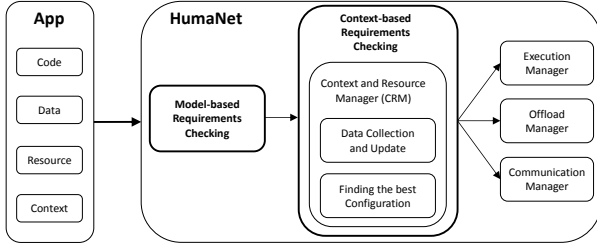


Fig. 3: Components of HumaNet architecture.

(Section IV-A2). The threshold will be used later to indicate whether the individual is mentally fatigue while driving.

III. HUMANET ARCHITECTURE

HumaNet provides an IoP framework that facilitates the communication and data aggregation among different nodes (apps and devices) for developing safe and sustainable apps. In this sense, every participating smartphone installs the middleware that runs in background. Aggregated data from group of people can be exploited to develop apps based on community inferences. On the other hand, to ensure safety and sustainability requirements, off-line model-based and run-time context-based checking are provided as seen in Figure 3.

A. Model-Based Requirements Checking

In model-based checking, models of a given app are developed and tested to analyze and optimize performance considering accuracy, response time, and energy efficiency requirements. Behavioral models are formed and studied to avoid unpredicted states of the app. Alongside, for optimization, the test results are assessed to improve the app configurations (e.g. sensors, algorithms, and platforms) through iterations.

B. Context-Based Requirements Checking

In this component, the IoP including sensors, apps, smartphones, and servers are monitored on-the-fly. The monitored data is analyzed to make system level decisions toward ensuring the requirements. Thus, HumaNet applies computational offloading (to a central server or smartphones in vicinity), controlling sampling rate of sensing modules, prioritizing execution of various apps, or adjusting trade-offs among accuracy, latency, and energy. For instance, the specifications (e.g. memory, workload, and battery level) of the candidate devices are checked to choose the best match for offloading. The functionalities for offloading are described below:

1) *Context and Resource Manager (CRM)*: The CRM module runs periodically within the device to ensure that the application meets specific context and resource requirements by sensing the context as well as the resources. The mobility of users, other running apps, and user's state can change the context and resource values of the mobile device. Based on the sensor information, the device context values are evaluated against the context and resource requirement of the app.

2) *Execution Manager*: This module manages the execution of the code. It downloads the code and data blocks assigned to it. The code is then run which takes input as the downloaded data and results are generated.

3) *Offload Manager*: Offload manager takes the inputs from the CRM to assess if the device currently meets the context specifications. In case the required context does not exist, the task is offloaded. Context and requirement specifications of the application tasks are considered to determine the type of task to be offloaded to the other devices.

4) *Communication Manager*: This module potentially offloads tasks to nearby available devices if needed.

IV. SAFEDRIVE IMPLEMENTATION

As mentioned in Section II, smartphone has a foreground application (SafeDrive client side) that computes mental fatigue levels and the collision detection algorithm (SafeDrive server side) is executed on the central sever. The HumaNet on the background is equipped with a timed automata model of the entire execution of SafeDrive. On estimation of a possible collision the time automata model is used to verify whether sending a warning message will lead to a safe condition. Using the timed automata model ensure that the driver reaches the desired mental alertness state much before the reaction time to the impending accident.

A. EEG-Based Mental Fatigue Detector Application

Electroencephalography (EEG) is the most commonly used technique for real-time analysis of brain signals to detect mental fatigue. The process of falling asleep is divided into three states, known as Awake, Non-Rapid Eye Movement sleep (NREM), and Rapid Eye Movement sleep (REM) [16], [17]. In each stage, EEG signals divulge specific characteristics. Table I lists main frequency bands extracted from EEG signals and their corresponding cognitive state. Mental fatigue can be recognized in the earliest stage of NREM. This stage coincides with diminution in power changes in alpha frequency band and simultaneously power raise in alpha and theta bands [16], [18].

1) *Mental Fatigue Detection Algorithm*: There are many algorithms to detect mental fatigue from EEG datasets including Mean Comparison Test (MCT) and Thresholding [19], Principal Component Analysis and Linear Regression Model [10], F-measure (the harmonic mean of precision and recall) [20], and support vector machine [11]. However, these algorithms work best on a desktop setting and require compute intensive operations. In our application, we use the Markov chain decision process, which according to our test results is computationally less expensive and has comparable accuracy.

TABLE I: EEG signals frequency bands and cognitive states.

Frequency Band	Cognitive State [16]
Delta (0.5-4 Hz)	Sleep Activity
Theta (4-8 Hz)	Attention Level
Alpha (8-13 Hz)	Relaxation at decreased Attention Levels
Beta (13-30 Hz)	Active Concentration and Alertness State
Gamma (30-100 Hz)	Perception

Research on mental fatigue detection shows that decrease in P_β/P_α and increase in $(P_\alpha + P_\theta)/P_\beta$ parameters expose mental fatigue clearly, where P_α , P_β , and P_θ indicate power level in Alpha, Beta, and Theta bands, respectively [18]. In our work, the Fast Fourier Transform (FFT) is performed on raw EEG signals to transform them from time to frequency domain. The signal amplitudes in the frequency domain are used to compute the average EEG powers P_α , P_β , and P_θ over 14 channels for Emotiv and one channel for MindWave. FFT is applied on every second of signal to update P_β/P_α and $(P_\alpha + P_\theta)/P_\beta$. Two thresholds (T_l , T_h) are determined which are proportional to the average value of P_β/P_α and $(P_\alpha + P_\theta)/P_\beta$ in alert state, respectively. The threshold calculation is described in Section IV-A2. We then employ a two-state Markov chain model (Figure 4) to compute the steady state probabilities (π_0 , π_1) for *alert* and *fatigue* states. The model parameters are defined as follow:

$$\begin{cases} d_{0,0} : P_\beta/P_\alpha > T_l \parallel (P_\alpha + P_\theta)/P_\beta < T_h \text{ in State 0} \\ d_{0,1} : P_\beta/P_\alpha \leq T_l \&\& (P_\alpha + P_\theta)/P_\beta \geq T_h \text{ in State 0} \\ d_{1,1} : P_\beta/P_\alpha \leq T_l \&\& (P_\alpha + P_\theta)/P_\beta \geq T_h \text{ in State 1} \\ d_{1,0} : P_\beta/P_\alpha > T_l \parallel (P_\alpha + P_\theta)/P_\beta < T_h \text{ in State 1} \end{cases} \quad (1)$$

π_0 and π_1 are updated every second, according to the parameters derived from the last 30s of data recording. As soon as π_1 exceeds beyond 50%, the condition of the user is interpreted as a mental fatigue state. When the fatigue state is detected, the application will make an alarm tone.

2) *Threshold Setting and Ground Truth*: We need to set thresholds accurately to detect mental fatigue in a timely manner and avoid false alarms as much as possible. According to sleep studies, human mental fatigue generally appears late at night and impairs performance [21]. Subsequently, in fatigue detection research, the most common way to determine thresholds on alpha, beta, and theta is through experiments in different times of a day [22], [23]. In these studies, they assume driver is alert in the morning and mentally fatigue late at night as their ground truth. We also take a similar approach.

Our experiment has two phases: *training* and *testing*. In the training phase, we need brain signal samples of a subject once she is completely alert and the other time when she feels drowsy. So, we asked the subject to wear the headset (e.g. Neurosky MindWave or Emotiv) for two 30-minute sessions; one in the morning and the other one late at night (signals are recorded with 512 Hz sampling rate). Signals from the morning session, are considered as alert state or baseline signals and vice versa. We applied MCT on the mean values of ratio indices P_β/P_α and $(P_\alpha + P_\theta)/P_\beta$ in each alert and mental fatigue state to set the thresholds. As a result, at least about 25% reduction in P_β/P_α value and about 10% growth in $(P_\alpha + P_\theta)/P_\beta$ value (compared to these values in alert state) indicates mental fatigue state. In the testing phase, the subject puts on the headset and run the mobile application in five trials on different days. Each trial has two 30-minute sessions, one in the morning and the other one late at night. The headset captures EEG signals and the smart-phone monitors, and processes them real-time. According to the threshold from the training phase, mental fatigue detection

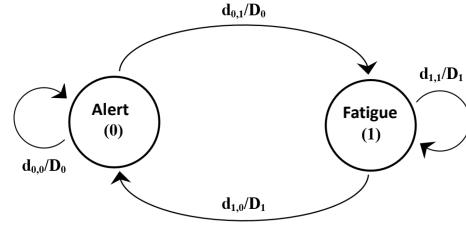


Fig. 4: The two-state Markov chain model. D_0 and D_1 represent the sums $d_{0,0} + d_{0,1}$ and $d_{1,0} + d_{1,1}$, respectively.

algorithm (Section IV-A1) can determine mental state (i.e. alert or fatigue) of the subject.

B. Vehicle Collision Detection Algorithm

In this algorithm (i.e. our previous work [24]), we assume that the transportation control system (Figure 1) can monitor the car parameters of a large number of cars on a highway or at least the speed of the vehicle precede and follow the vehicle whose mental fatigue is being monitored. To predict the motion of the concerned vehicle, lateral and longitudinal control algorithms [25] can be used. The lateral control algorithm generates steering angle (i.e lateral control angle) based on the current position and the next destination over a short amount of time as given by the following equation:

$$\delta = \arctan[2l(3y - x \tan \theta)/x^2], \quad (2)$$

where δ is the steering angle, l is the wheel base of the vehicle, (x, y) is the next way point, and θ is the heading angle with respect to the road. The longitudinal control algorithm generates speed based on the preceding and following vehicle's speed as per the equation given below:

$$v_r = v_p + k1(v_p - v_f) + k2(L_r - L_m) \quad (3)$$

where v_r is the speed of the AV, v_p and v_f are the speed of the preceding vehicle and following vehicle respectively, L_r is the minimum longitudinal distance between two vehicles, L_m is the measured inter vehicular distance, $k1 = m1\|L_r - L_m\|/\|L_r\|$, $k2 = m2k1$, and $m1$ and $m2$ are control gains.

A collision can be detected whenever either -

a) $v_r - v_p > \sqrt{(x_r - x_p)^2 + (y_r - y_p)^2}/\tau$, where (x_r, y_r) and (x_p, y_p) are the positions of the vehicle and the preceding one, respectively and τ is a small amount of time that is less than the typical reaction time of human drivers, road conditions, and speed limits.

b) $v_f - v_r > \sqrt{(x_f - x_r)^2 + (y_f - y_r)^2}/\tau$, where (x_r, y_r) is the current position of the following vehicle, or

c) $\sqrt{(x - x_p)^2 + (y - y_p)^2} \leq \epsilon$ or $\sqrt{(x - x_f)^2 + (y - y_f)^2} \leq \epsilon$, where ϵ is a parameter determined by the road conditions and highway speed limits.

C. Modeling Real-Time Safety Properties

In the SafeDrive application, temporal constraints can limit the choices of mental fatigue detection algorithm that can be used in the wearable devices. The computation discussed in Section IV-B can accurately predict imminent collision 6s in

the future. Within this time interval the smartphone has to compute mental fatigue levels and based on the thresholds, the smartphone will decide to set the alarm. The alarm has to be set at a time such that the driver has enough time to maneuver out of the dangerous situation. The reaction time of a driver in highway with a speed limit of 65 mph is around 2-4s [26], [27]. This leaves the mental fatigue application only 2s to make a decision.

Figure 5 shows a Finite State Automata (FSA) representation of the SafeDrive application. The FSA has states and transitions between them. Each transition operates on some input $x = x_i$ and assigns an output $y := y_i$ and the transition condition is denoted by $x = x_i/y := y_i$. The FSA for SafeDrive app has five states, four inputs (X , Z , T , and P), and two outputs (Y and C) as described in Figure 5. The FSA in each vehicle starts in the state E. When a car enters the danger zone $X = 1$, it sets $Y = 0$ and sends baseline EEG activity to the server and the FSA transits to state L. In the state L if there is an threshold update request then $T = 1$ and the FSA goes to state U and after the update comes back to L. If the collision detection algorithm detects collision then $Z = 1$ and the FSA goes to the state D to invoke the driver drowsiness alertness algorithm. If the driver is alert then $P = 0$ and $Y = 1$ since the driver sends the drowsiness levels to the server. If the driver is drowsy then the FSA goes to alarm (A) state and sets $Y = 1$. In the alarm state if there is a collision then $C = 1$ and updated to the server. Else when the driver gets out of the danger zone, it updates the server with the values for Y and C and exits. Given this FSA specification real-time properties can be specified using a combination of temporal logic and timer clock in the FSA. The temporal properties include: a) *update requirement*: after every new update threshold message from the client the brain server has to immediately update its software, and b) *collision avoidance window*: since the collision predictor can only predict 6s in future, and driver reaction time in the worst case is 4s, the entire process including collision data availability, mental fatigue detection, and alarm setting has to be done within a window of 2s.

In Linear Temporal Logic (LTL), a proposition $x \implies y$ denotes the logical function $\neg x \vee (x \wedge y)$. Some of the key operators used in this paper are Γ , Υ , and ξ . The operator Γ applied on a proposition M denotes that the proposition is true for any sequence or subsequence of states in the finite state automata. This effectively means that it is true for each state of the FSA. The operator Υ applied on a proposition M denotes that the proposition eventually be true for some time $t > 0$. The operator ξ applied on a proposition M denotes that the proposition is true in the next state.

The proposition L , D , U , E , or A are true if the FSA is in the corresponding states. The update requirement can be expressed using the formulation $\Gamma(L \wedge T = 1 \implies \xi U)$. The collision avoidance window can be expressed by introducing another time variable in the FSA. In each state of the FSA, we include the equation $t = t + \tau$, where τ is the time resolution of the FSA. In the state L , we consider that whenever $Z = 1$

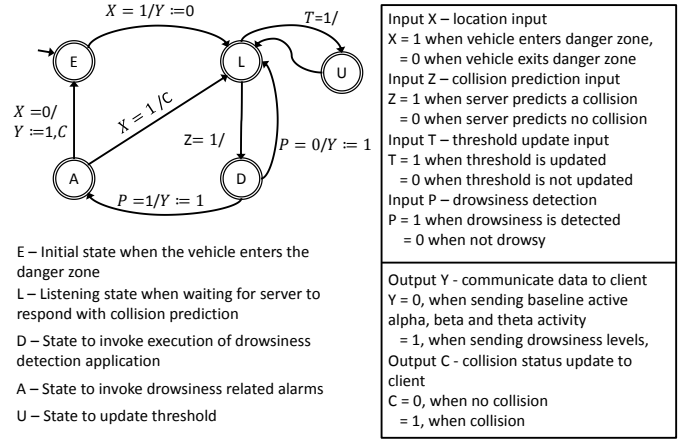


Fig. 5: Finite state automata representation of the SafeDrive.

we set $t = 0.2s$, which is the execution time of the prediction algorithm on the client side and the communication delay. Now the requirement is that there has to be a mental fatigue related alarm if $P = 1$ within $t = 2s$ can be expressed as follows -

$$\Gamma(L \wedge Z = 1 \implies (XD \wedge \Upsilon(P = 1 \implies (\Upsilon A \wedge (t < 2.0s))))))$$

V. EVALUATION OF SAFEDRIVE

We evaluate SafeDrive application implementation using HumaNet with respect to two metrics: a) real-time safety requirements, and b) energy and power consumption.

A. Real-Time Safety Requirements

For verifying the safety requirement, we need to consider accuracy and response time of our algorithm.

1) *Accuracy*: The True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) were calculated for each trial, to obtain: Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$, Sensitivity = $\frac{TP}{TP+FN}$, and Specificity = $\frac{TN}{FP+TN}$ as listed in Table II. In these formulas, true positive represents the number of alarms in fatigue state, and false negative is equal to number of alarm misses in fatigue state. On the other hand, true negative is the number of alertness detection in alert state, and false positive represents the number of alarms in alert state. Test results in Table II show that the application can detect mental fatigue with 91% accuracy using Emotiv before the user fall asleep and completely lose control of the vehicle. MindWave has just one electrode placed on the user's forehead. So, compared with to Emotiv, which uses 14 channels, more test sessions are required using MindWave to set up thresholds and reach the accuracy of Emotiv.

2) *Response Time*: We measured the end-to-end latency of the SafeDrive app. For Emotiv, the average end-to-end latency of the system is 2.031s with variance (σ^2) = 0.142 and standard deviation (σ) = 0.377. For Neurosky, the average end-to-end latency of the system is 0.995s with $\sigma^2 = 0.223$ and $\sigma = 0.472$. According to collision studies in highways (Section III-A), with these response times, the system can avoid accidents caused by mental fatigue of the driver and satisfy real-time expectations of the system.

TABLE II: The experimental results in five test trials.

Test Trial	1	2	3	4	5	Average
Accuracy	93%	92%	88%	91%	95%	91%
Sensitivity	86%	85%	76%	81%	90%	83%
Specificity	100%	100%	100%	99%	99%	99%

B. Energy and Power Consumption

One of the key concerns in designing any practical, efficient, and sustainable systems is to keep the amount of energy and power consumption as low as possible. By using Emotiv, a rough estimation of battery usage shows that when the application is running on the phone, we have 2% more battery discharge. Power monitoring on the phone indicates that the running application consumes about 130 mW on average. Monitoring battery usage using MindWave for EEG recording shows a 1% increase when the application runs on the phone and consumes about 109 mW power on average. There is an inverse relation between power and latency. We can decrease power usage by reducing frequency and increasing response time. So, the trade off between power and time should be considered for real-time processes.

VI. CONCLUSIONS

In this paper, we introduce HumaNet architecture for developing real-time driving assistant mobile applications for aware cities. HumaNet enables apps using domain specific knowledge, to rapidly be implemented in a community networking framework that is optimized for accuracy, response time, and energy efficiency. HumaNet allows apps to share data and hence reduces cost of physiological sensing. We implement a prototype version of HumaNet and show its usage for developing the SafeDrive application. SafeDrive application requires sharing of physiological data among individuals and provides accurate physiological feedback through auditory stimulation in real-time. In this sense, HumaNet platform applies model-based and context-based optimization techniques to satisfy safety and energy efficiency requirements in SafeDrive.

ACKNOWLEDGMENTS

This work has been partly funded by CNS grant #1218505, IIS grant #1116385, and NIH grant #EB019202.

REFERENCES

- [1] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter cities and their innovation challenges," *Computer*, vol. 44, no. 6, pp. 32–39, 2011.
- [2] J. M. Schleicher, M. Vögler, C. Inzinger, and S. Dustdar, "Towards the internet of cities: A research roadmap for next-generation smart cities," in *Proceedings of the ACM First International Workshop on Understanding the City with Urban Informatics*. ACM, 2015, pp. 3–6.
- [3] A. Mostashari, F. Arnold, M. Maurer, and J. Wade, "Citizens as sensors: The cognitive city paradigm," in *Emerging Technologies for a Smarter World, 8th International Conference & Expo on*. IEEE, 2011, pp. 1–5.
- [4] K. S. Oskooyee, A. Banerjee, and S. K. S. Gupta, "Neuro movie theatre: A real-time internet-of-people based mobile application," in *The 16th ACM International Workshop on Mobile Computing Systems and Applications*. ACM, 2015.
- [5] J. Sohankar, K. Sadeghi, A. Banerjee, and S. K. S. Gupta, "E-bias: A pervasive eeg-based identification and authentication system," in *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. ACM, 2015, pp. 165–172.
- [6] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power, "Using social media to enhance emergency situation awareness," *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 52–59, 2012.
- [7] B.-G. Lee and W.-Y. Chung, "Driver alertness monitoring using fusion of facial features and bio-signals," *Sensors Journal, IEEE*, vol. 12, no. 7, pp. 2416–2422, 2012.
- [8] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, "Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 1, pp. 121–131, 2011.
- [9] M. M. Bundele and R. Banerjee, "Detection of fatigue of vehicular driver using skin conductance and oximetry pulse: a neural network approach," in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*. ACM, 2009, pp. 739–744.
- [10] C.-T. Lin and et. al, "Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning," *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 5, pp. 1582–1591, 2008.
- [11] I. Shin and et. al, "Development of drowsiness detection system with analyzing attention and meditation wave using support vector machine method," *ISICO 2013*, 2013.
- [12] "Neurosky body and mind quantified. neurosky.com."
- [13] A. Stopczynski, C. Stahlhut, J. E. Larsen, M. K. Petersen, and L. K. Hansen, "The smartphone brain scanner: a portable real-time neuroimaging system," *PloS one*, vol. 9, no. 2, 2014.
- [14] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 283–299, 2012.
- [15] M. Pore, K. Sadeghi, V. Chakati, A. Banerjee, and S. K. S. Gupta, "Enabling real-time collaborative brain-mobile interactive applications on volunteer mobile devices," in *Proceedings of the 2nd International Workshop on Hot Topics in Wireless*. ACM, 2015, pp. 46–50.
- [16] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting driver drowsiness based on sensors: a review," *Sensors*, vol. 12, no. 12, pp. 16937–16953, 2012.
- [17] P. A. Bryant, J. Trinder, and N. Curtis, "Sick and tired: does sleep have a vital role in the immune system?" *Nature Reviews Immunology*, vol. 4, no. 6, pp. 457–467, 2004.
- [18] H. J. Eoh, M. K. Chung, and S.-H. Kim, "Electroencephalographic study of drowsiness in simulated driving with sleep deprivation," *International Journal of Industrial Ergonomics*, vol. 35, no. 4, pp. 307–320, 2005.
- [19] J. Park, L. Xu, V. Sridhar, M. Chi, and G. Cauwenberghs, "Wireless dry eeg for drowsiness detection," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3298–3301.
- [20] C.-T. Lin and et. al, "A real-time wireless brain-computer interface system for drowsiness detection," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 4, no. 4, pp. 214–222, 2010.
- [21] C.-T. Lin, R.-C. Wu, T.-P. Jung, S.-F. Liang, and T.-Y. Huang, "Estimating driving performance based on eeg spectrum analysis," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 3165–3174, 2005.
- [22] T. Kurita, N. Otsu, and N. Abdelmalek, "Maximum likelihood thresholding based on population mixture models," *Pattern Recognition*, vol. 25, no. 10, pp. 1231 – 1240, 1992.
- [23] S. H. Kwon, "Threshold selection based on cluster analysis," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1045 – 1050, 2004.
- [24] S. Kandula, T. Mukherjee, and S. K. S. Gupta, "Toward autonomous vehicle safety verification from mobile cyber-physical systems perspective," *SIGBED Rev.*, vol. 8, no. 2, pp. 19–22, jun 2011.
- [25] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, no. 3, pp. 155 – 161, sep. 2002.
- [26] A. Mehmood and S. M. Easa, "Modeling reaction time in car-following behaviour based on human factors," *International Journal of Applied Science, Engineering and Technology*, 2009.
- [27] J. W. Muttart, "Quantifying driver response times based upon research and real life data," in *3rd International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, vol. 3, 2005, pp. 8–29.