2020

Modeling, Designing and Applying Machine Learning Algorithms for Driver Drowsiness Detection

Mohsen Babaeian
Claremont Graduate University

Modeling, Designing and Applying Machine Learning Algorithms for Driver Drowsiness Detection

By

Mohsen Babaeian

Claremont Graduate University

2020

Approval of the Dissertation Committee

This dissertation has been duly read, reviewed, and critiqued by the Committee listed below, which hereby approves the manuscript of Mohsen Babaeian as fulfilling the scope and quality requirements for meriting the degree of Doctor of Philosophy in Engineering & Industrial Applied Mathematics.

Mohammad Mozumdar, Chair

Claremont Graduate University

Associate Professor of Electrical Engineering

Allon Percus, Professor

Claremont Graduate University

Associate Professor of Mathematics

Ali Nadim

Claremont Graduate University

Associate Professor of Mathematics

Henry Yeh, Professor

California State University Long Beach

Associate Professor of Electrical Engineering

Abstract

Modeling, Designing and Applying Machine Learning Algorithms for Driver Drowsiness Detection

By

Mohsen Babaeian

Claremont Graduate University: 2020

Driver drowsiness has been a significant hazard resulting in various traffic accidents. Therefore, monitoring this condition is crucial not only in alerting drivers, but also in avoiding fatal accidents. Many research studies propose new systems to reduce the number of drowsiness-related injuries and fatalities. The ultimate goal for a drowsiness detection system is to detect the drowsiness on time and minimize the system or environment errors to avoid false readings, such as studying physiological signal processing patterns. These potentially life-saving systems must operate in a timely manner with the highest precision. Researchers proposed various methods based on driving pattern changes, driver body position, and physiological signal processing patterns. There is a focus on human physiological signals, specifically the electrical signals from the heart and brain. In this study, we are presenting an alternative method to determine and quantify the driver drowsiness levels using a physiological signal that was collected in a non-intrusive method. This methodology utilizes heart rate variation (HRV), electrocardiogram (ECG), and machine learning for drowsiness detection. It is apparent that a driver’s drowsiness is associated with an immediate change in heart rate, and due to the fact that Electrocardiogram (ECG) is used to detect an accurate heart rate. We used it as a parameter in the proposed design where it consists of a non-contact ECG sensor as an input source and a circuit with a two-stage amplifier to improve the ECG signal's strength and filters to minimize noise. An approximate maximum peak ECG output voltage of 2.8V was obtained in LT Spice, and the resulting ECG output is sufficient enough to detect a driver's drowsiness while preventing major accidents. Furthermore, the HRV is measured with an ECG. The algorithm uses both wavelet and short Fourier transform (STFT). The algorithm extracts and selects the desired features. Then, the system applies both the support vector machine (SVM) and K-nearest neighbor (KNN) method. This achieves an accuracy of 80% or higher. In this research, the accuracy output for the SVM method is 83.8%, 82.5% when using STFT, and 87.5% when applying the WT technique. The algorithm with highest accuracy helps to decrease the number of accidents due to drowsiness. Furthermore, we applied unsupervised machine learning (clustering) to study the behavior of HRV during drowsiness. We can measure different levels of drowsiness based on the changes in the density and shape of the HRV clusters by using this method. Moreover, the pre-measured labeled data is not required to establish the algorithm in this method. Therefore, this algorithm evaluates drowsiness and no prerecorded data is required for any unknown object or person. Successful application of this drowsiness detection method may help to avoid traffic accidents. This study may be beneficial for policy maker's in preparing regulations to prevent traffic accidents worldwide and may also helpful for users to increase their knowledge and awareness regarding drowsiness detection.

Keywords: Drowsiness, Machine Learning, Electrocardiogram (ECG), Heart Rate Variability (HRV), Wavelet Transform (WT).

Table of Contents

Index of Figures

Index of Tables

Introduction

Recently, the rate of traffic accidents has increased in conjunction with the increase in vehicle numbers. Among the array of diverse automobile accident scenarios, driver drowsiness is one of the most dangerous situations; similar to alcohol or drugs, driver drowsiness, which can be caused by fatigue, extensive driving, sleep deprivation, a low circadian rhythm, or medication usage. The risk of driver drowsiness increases with prolonged driving, lack of sleep, and alcohol consumption. Drowsy driving contributes to traffic accidents by delaying the driver’s reaction time. Therefore, traffic accidents related to driver drowsiness are major safety issues. World Health Organization (2015) reported that over 1.5 million people die per year and over 40 million people have severe injuries resulting from driver drowsiness related accidents. The National highway traffic safety administration (2017) reported the total 411 deaths due to drowsy driving between 2013 and 2017. Moreover, it estimates the total motor vehicle crashes involving drowsy driving around 91,000 accidents and also over 50,000 injuries in 2017. However, NHTSA (2018) reported a 2.4 percent decrease in fatalities overall in year 2018.

To prevent loss of human lives due to driver drowsiness and also lower the financial impacts, a precise system that detects drowsiness accurately and alerts drivers against hazards is highly desired. Drowsiness is interpreted as the transition from alertness to rapid eye movement (REM) and from non-rapid eye movement (NREM) to alertness and vice versa. Usually, the drowsiness stage is detected when a subject’s body movement slows significantly. The sleep cycle consists of three major periods: awake, non-rapid eye movement (NREM), and rapid eye movement (REM). REM consists of deep sleep and vivid dreaming. On the other hand, NREM represents light sleep, just after falling asleep. NREM is classified into three more stages namely as N1, N2 and N3. In stage N1, a person gets drowsy and drowsing into sleep. The main focus of this research is to detect this stage in the earliest time. The N1 and N2 stages are deeper sleep and delta waves propagated from human brain are seen in these stages. If a researcher measures the transition time from awake to N1 for a subject, this measurement is a unique number for that subject. Xun (2012) stated that the level of drowsiness is evaluated by analysis and it varies from person to person. Yilmaz, Asyali, Arikan, Yetkin, and Ozgen (2010) stated that there are many different technical implementations that evaluate these levels and convert them to measurements.

Due to the fact that driver drowsiness is a rising challenge in automobile industry, driver safety has been one of the leading researches by the researchers. Several methods and techniques have been proposed for drowsiness detection throughout the years due to the prevalence of driver drowsiness related incidents. Langille (2019) proposed that driver's drowsiness is detected through various approaches such as analyzing a driver's physical behavior, vehicle response, brain waves, pulse rate, and respiration. Furman and Baharav (2010) stated that various methods have been developed using sensors detecting several physiological signals that were measured through electroencephalogram (EEG), electrooculogram (EOG), and electrocardiogram (ECG) in order to detect drowsiness in its early stages. Borghini, Astolfi, Vecchiato, Mattia and Babiloni (2014) explained that drowsiness is associated with the physiological signals; these measurements provide accurate results that are based on the correlation between physiological signals and sleep. Because the driver's body automatically generates physiological signals and he/she has no control to alter them, the physiological signal detection methods have certain advantages over body movement pattern detection techniques such as eyelid movement. Sztajzel (2004) stated that Electrocardiogram (ECG) signals are one of the most interesting topics of driver drowsiness. Muhamad, Nasreen and Michael (2017) stated that the ECG analysis method proves more proficient due to its reliance on Heart Rate (HR), among the array of physiological signal analysis methods used for detecting driver drowsiness. Sahaydhas, Sundaraj and Murugappan (2012) have suggested a hybrid driver drowsiness detection system that utilizes multiple sensors (i.e. a strain gauge sensor and ECG sensors). They proposed two methods to determine a driver's drowsiness. Initially, it is detected using a verbal questionnaire. Drowsiness levels are evaluated using the driver's response and several tools has been used to convert this rating to measure a driver's drowsiness. Secondly, the change in the sensed data is analyzed by utilizing sensors to measure drowsiness. Thomas, Andreas and Nikoletta (2017) stated that the limitation of this system is verbal questionnaire that cannot be implemented during driving. Swapnil, Bharti, Ansari and Khatib (2014) suggested a driver's fatigue detection technique using a strain gauge sensor; this system consists of a sensor, a signal processing module, and an alarm-producing device. The sensor is positioned in front of the driver and monitors eye and jaw movement's micro-sleeps and obtains input data. As soon as the driver feels drowsy, fatigue is detected, and the system produces an alarm to alert the driver, but while this technique is innocuous, the design is costly and requires complex processing techniques. Ghosh, Nandy and Manna(2015) presented a computer vision system that uses eye tracking to monitor driver drowsiness in real-time; the system consists of a camera that captures images of the face, a data acquisition block that implements algorithms for face, eye, and pupil detection, and a processor that analyzes drowsiness from eye movements. However, the main drawback of this method is a change in the eye's sensitivity, which produces false positives in the results. Assari and Rahmati (2011) proposed a system consisting of a camera and an infrared LED, and by using infrared LEDs, it was able to overcome the issue of intensity variation that resulted from external car lights. Nevertheless, the main shortcoming of this technique is that it yields fault results. For example, the system will determine a person is drowsy from his appearance, when he may not be in reality.

Kumar, Samuel and Santhosh (2012) suggested a drowsiness detection system using electroencephalogram (EEG) signals. It consists of an EEG detection circuit which comprises an EEG sensor that detects and amplifies the tiny electrical voltages generated by brain cells, a micro-control unit that produces a control signal for processing detected EEG signals and an EEG signal processing circuit that processes the EEG signals to identify the driver's drowsiness. The EEG sensors used in this design are wired. Hence, it is not suggestible that they can be used while driving, because it impairs driver's concentration. Deepa and Vandana (2016) proposed a system for drowsiness detection using EEG signals. It contains an EEG sensor that detects EEG signals from the brain, an EEG signal acquisition unit, which amplifies and filters the signal for drowsiness analysis, and a mobile unit that alerts the driver if drowsiness related information is observed in the sensed EEG signals. Laguna, Moody and Mark (1998) stated that it is necessary to note that the wired EEG sensors used in this proposed system are not safe to implement as it affects the driver's concentration. Sang-Joong, Heung-Sub and Wan-young (2014) presented a new ECG sensor with conductive fabric electrodes, which can be positioned on a car's steering wheel in order to identify a driver's drowsiness, and the measured ECG signal from the driver's palm has a sampling rate of 100HZ. However, the sensors utilized in this design are complicated to employ because the skin-electrode impedance may result in poor quality of the ECG signal. Xun (2009) suggested a driver's drowsiness system using two non-intrusive conductive fabric ECG sensors placed on the steering wheel and the back of the driver's seat. The design consists of signal conditioning circuitry such as a notch filter, a band pass filter, an amplifier and a driven right-hand circuit for improving the strength of the ECG signal obtained from two different sensors. He implemented an adaptive filter algorithm in the software to reduce the baseline noise in his proposal. Although, it is necessary to note that sensors used in this method will fail to sense the ECG data, if a person is wearing any gloves or if the driver uses only one hand. Sangeetha, Rajendiran and Malathi (2015) presented an embedded driver drowsiness detection system that not only monitored and controlled the drowsiness state, but also provided feedback to stop the automobile once drowsiness was identified. The input obtained from an ECG sensor is amplified and processed to determine the driver’s state, and if the driver is drowsy, the processor notifies the driver by emitting an alarm and activating the driver circuit.

The ECG signal is a periodic electrical signal that consists of three major parts, QRS complex, P wave, and T wave (Figure1). The definition of HRV is a time series that measures the time distance between consecutive R peaks. R peaks refers the maximum amplitude in the R wave. An accurate R peak detection is very essential in signal processing equipment for heart rate measurement. Lee, Lee, Sim, Kim and Park (2009) proposed the usage of non-contact sensors to detect drowsiness and the implementation of an algorithm to trace energy changes in driver’s heart rate variation (HRV). Li and Chung (2013) stated that HRV signal is correlated to the responses of the autonomic nervous system (ANS) to sympathetic and parasympathetic variations. Since the Autonomic Nervous System (ANS)regulates sleep and adjust heart rate functionality. Therefore, analyzing HRV is an accurate and non-invasive way to assess drowsiness. They suggested that using non-contact sensors to detect drowsiness by measuring HRV frequency responses. They indicated that the ratio of the two highest energy regions in the frequency domain change in proportion to the level of drowsiness. Noise has a negative impact on the system’s output accuracy, and it is highly difficult to eliminate noise when using non-contact sensors. The disruptive, mobile environment of a car absorbs noise interference, and in removing the background noise in post-processing, errors can occur if any relevant data is lost in the transition. Noise can negatively and severely impact the output of the reading if it is not eliminated during processing. The sources of noise vary and include environmental noise, power line noise, hardware system noise and anomalous data. Chipman and Jin (2009) stated that the environment noise interference also alters the level of drowsiness. To develop a system with minimized detection time and a maximized accuracy, it is imperative to remove all noise and anomalous data to minimize the processing time. Therefore, a dynamic algorithm constantly updates the level of drowsiness measurement for each person and does not require any pre-recorded data. At the same time, it will increase the processing time since it updates itself continuously.

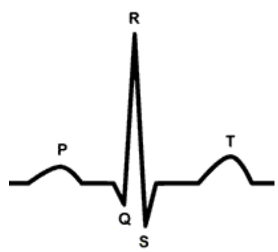

Figure 1. Electrocardiogram Signal

The early drowsiness detection decreases the possibility of the accident. The accuracy of drowsiness detection prevents any false reading. Therefore, the most significant challenge while developing this system is maintaining high accuracy while minimizing the detection time. Noise removal techniques deliver a clean ECG signal to the processor and increase the system's accuracy while minimizing its detection time. After extracting the time series and then the frequency response, it is evident that there are four areas that contain concentrated energy levels: ultra-low frequency (ULF: 0.0001-0.001 Hz), very low frequency (VLF: 0.004-0.02 Hz), low frequency (LF: 0.06-0.12 Hz), and high frequency (HF: 0.15-0.4 Hz). The energy ratio of LF to HF is known to change consistently as an individual gets drowsier. The change in this ratio is used as an indicator of drowsiness. To improve accuracy and minimize time detection and noise, the machine learning (ML) technique were incorporated into the system and the results were studied and published.

In the previous studies, Khazaee and Ebrahimzadeh (2010) suggested a new method to extract features in frequency domains using fixed-sized windows and selecting the support vector machine (SVM) classifier for the generalization process. It depicted an SVM classifier with a Gaussian rule-based fuzzy (GRBF) kernel which achieved over 95% accuracy if the noise interference from the training and test data is correlated. Otherwise, the system's accuracy may plummet to 70 percent. Zhao and Zhan (2005) tested a method for feature extraction by using wavelet transform and applying SVM with a Gaussian kernel, but while the proposed approach attained an overall 98% accuracy, the method's complexity had a slow processing time. Babaeian, Bhardwaj, Esquival and Mozumdar (2016) formulated a new approach by sliding a fixed window along the signal to extract features by applying a low complex Logistic Regression (LR) classifier. These researchers aimed to minimize the time detection and increase accuracy with the new techniques described in methods. The new algorithm incorporates ensemble logistic regression (ELR) with the regularization technique to overcome the overfitting problem and also employs wavelet technique over short-time Fourier transform (STFT) to adopt the non-stationary nature of ECG. Due to the low complexity of the LR technique, the algorithm's speed increases. However, if the number of features exceeds the number of instances, the LR algorithm is more likely to be overfitted. In that case, the algorithm could have a low accuracy with different ECG data. They proposed a new algorithm based on sliding and damping a window along a stream of temporal data and correlation between these windows statistics over time. Wu et al. (2007) categorizes the sleep stages sleep as rapid eye movement (REM) and non-rapid eye movement (NREM) and classifies the HRV data with random forest technique and achieved 88% accuracy. However, this method needs high number of features in frequency and time domain and automatically, it will increase the chance of over fitting.

Due to the fact that drowsiness variation is correlated with the HRV frequency domain parameters, we implemented few techniques to how to apply different frequency applications such as Short Fourier Transform (STFT) and Wavelet, to improve the data analysis in frequency domain and also, applying different techniques to improve the final results. In the next chapter, we simply applied logistic regression technique along with STFT and compare with other machine learning technique such as naïve Bayes and compared the outlet. Furthermore, we improve the model by applying wavelet technique and adding Ensemble Logistic Regression (ELR) and also, we applied Support Vector Machine(SVM) and K- Nearest Neighbor methods and compare the results and advantages of each techniques compare to the others.

Raudys and Jain (1991) explain the effects of a small size training set for pattern recognition and how it affects the system accuracy. They discussed the effect of high dimensionality (number of features) and choosing the window size for density estimation. Jain and Zongker (1997) measured the effect of extracting a high number of features from small size data and measure and compare the overfit effects for the small training sets.

We developed a novel method by applying a new clustering technique to eliminate the need of offline processing during the real-time measurement of drowsiness. If there is no pre-existing ECG training set, then the algorithm would not know whether the person will be either asleep or awake at the time of testing. The raw ECG data does not have any information with which it can determine the level of drowsiness. Therefore, the data is unlabeled and unsupervised machine learning is suggested at this point. Furthermore, collecting pre-recorded data for each subject is very difficult, since the data has to be collected in two states: drowsy and alert, but it is very difficult to collect such data, especially for drowsiness. Moreover, the drowsiness level changes from person to person. Even in a single subject, the drowsiness level varies over time. For example, as a person gets older, his or her drowsiness level changes.

1. Chapter1. Hardware and Software Design

1.1 Proposed System

In this section, we will describe our proposed hardware to detect drowsiness. The system is based on heart rate variation, time series detection and extracting. According to Raudys et al. (1991), frequency domain consists of four major intervals, (Ultra-low frequency, very low frequency, low frequency and high frequency). The low to high frequency ratio drops with drowsiness as described by Park et al. (2004). To extract frequency spectrum from a time series using Short Fourier Transform (STFT) is a timely process and accuracy is affected by noise and the algorithm accuracy. Due to the importance of driver mobility, a non-intrusive drowsiness detection system is required. Our proposed system is broken down into three main parts namely as hardware infrastructure, software designs and algorithm development.

1.1.1 Hardware Architecture

In the medical field, ECG is a type of electrical signal derived from automatic heart chemical recombination; generating different electrical potential voltages as stated by Silva et al. (2013). Usually, these electrical signals are detected with electrodes directly attached to a person's chest. However, it is important that the driver have to maintain his or her freedom or mobility and have no any attached sensors or wiring while driving. In this regard, Ghosh et al. (2015) suggests placing non-contact ECG sensors on either the seat back or the seat belt of the driver. Additionally, the driver's clothing will be between the sensors and the driver's skin. Therefore, various types and thicknesses of material will add undesirable noise to the ECG signal. Kuronen (2013) stated that it is imperative to remove all noise from the ECG signal and derive accurate information from the clean ECG signal in order to analyze the signal, only from the data actually generated by the driver's heartbeat. The output from the aforementioned non-contact sensors is a low voltage (mV) signal. Amplifying the signal is necessary for extracting information. Furthermore, Ghosh et al. (2015) suggests that a differential amplifier with two non-contact sensors will increase the gain as well as add a feedback technique (DRL). In both ways, the noise effects can be reduced. The feedback technique is implemented by inverting the output of the differential amplifier and connecting it to the driver's body. Also, noise generated by power lines that run alongside some road and major highways must be removed with a notch filter. Since the sensors are non-contact, they generate a lot of noise in a wide range of frequencies, from fractions of a Hertz (Hz) up to one hundred Hz as described by Sangeetha et al. (2015). In this study, we remove both the higher noise frequencies and the DC noise components with hardware filtering. Figure 2 shows different hardware stages in our implementation. He suggests that using 510 times voltage amplification, five high pass filters, and a 35 Hz low pass filter to remove a majority of this noise. The placement of sensors takes place around the body of a driver. Different sensor placement around the driver's body generates different levels of noise due to the distance from heart.

In the next step, a combination of micro-controllers and A/D convertors generates a discrete ECG signal and prepares the data for the software stage. Since the signal is low frequency, it is better to use a processor with a low frequency sampling rate. In this experiment, two A/D converters are used. An Arduino with a 10 kHz sampling rate and a Beagle Bone Black with a 200 kHz sampling rate is tested. However, using a higher frequency increase the memory storage size.

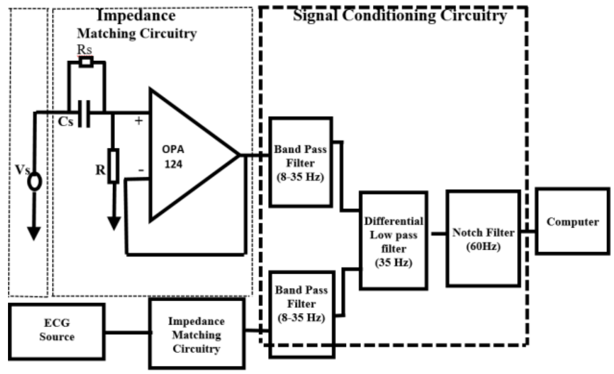

Figure2. Hardware design

1.1.2 Software Design

The primary focus of software development is to calculate the time series between every two consecutive R complex peaks and then store this data. However, if environmental noise still exists after this process while detecting peak for QRS complex, particularly in the same HRV frequency range, then the software can pick out a wrong peak (figure 3).Hardware filters help to remove noise in higher frequency ranges and the DC component. However, noise in the same frequency range as HRV survives in the hardware filtering stage. Pagani et al. (1986)stated that any noise would distort the ECG signal and can affect the peak to peak (RR) readings, erroneously detecting wrong peaks. For example, the S complex could be lifted up and then the next peak reading would be an S complex peak.

In order to clean the ECG signal, we chose the least means square (LMS) adaptive filter method with an LMS algorithm. This method removes the noise within the same frequency range which is mixed with the ECG signal. This algorithm's digital Finite Impulse Response (FIR) filter with adjustable weights filters out all noises with the deepest gradient method. In figure 3, there are two inputs, discrete function s(n) plus noise n(k) and correlated noise n(k).

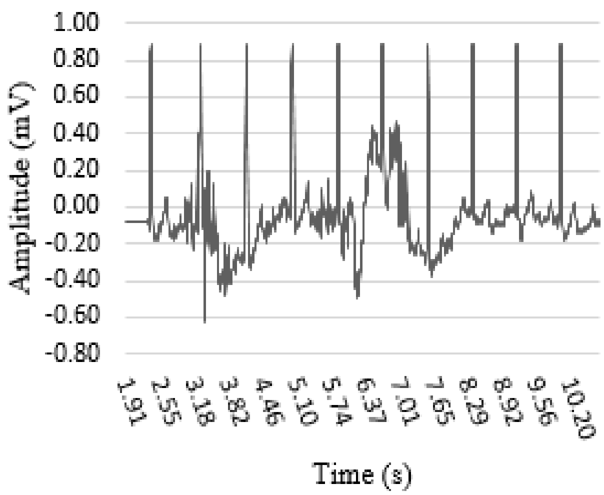

Figure3. ECG signal squared

The FIR filter adjusts the coefficient filter by applying the deepest gradient method. This method maximizes the correlation between the noise-contaminated ECG signal and environmental noise. The coefficients of the adaptive filter vary with noise which also changes over time. However, we expect a clean ECG signal in e(k) output. LMS algorithm has many advantages over other algorithms such as fast processing, since it does not require any squaring, averaging or differentiating when it minimizes the error in the steepest gradient method.

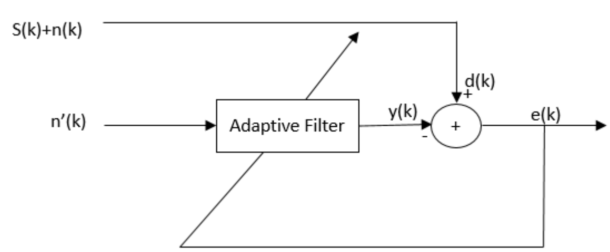

Figure4. Adaptive filter (LMS algorithm)

Reading peak-to-peak signals generate a time series with unevenly distributed output over time as shown in figure 5. Sang-Joong et al. (2014) proposed a method to increase the accuracy of peak-to-peak reading, the ECG signal is squared which increases R amplitude higher than the other complexes and make it easier to detect R peaks. Due to the fact that heartbeat doesn't follow a regular pattern, the reading is unevenly distributed over time. It is recommended to resample non-uniformity sampled signal. Patela, Lala, Kavanagha and Rossiterb (2010) stated that the data set is re-estimated in equal intervals in the resampling process. Furthermore, Kuo and Lee (2001) stated that the resampling method is tied to the Monte Carlo technique in which simulation estimates new data with minimum error.

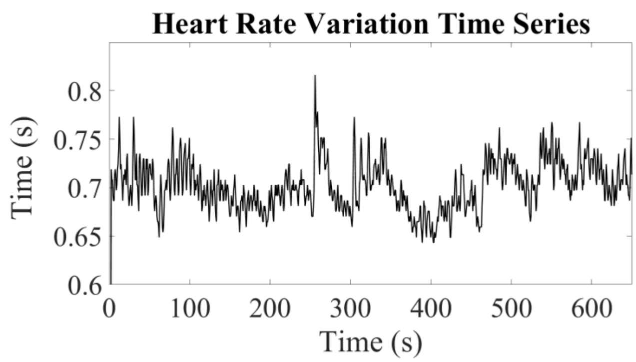

Figure 5. Peak-to-peak ECG time series

1.2 Design and Implementation

1.2.1 Design of the System

A detailed explanation of the proposed architecture is shown in figure 6.

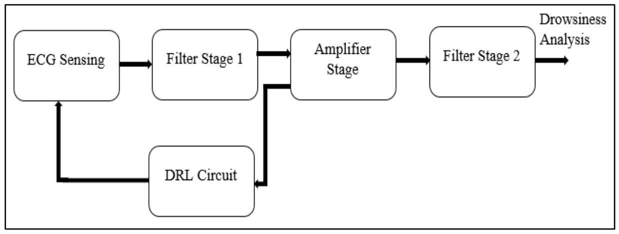

Figure6. Drowsiness detection system architecture

i. **ECG Sensing:**Traditionally, ECG devices have had electrodes, which are placed on the patient's skin in order to determine the heart's electrical activity. Recently, the use of ECG signals has extended into the technological field. Initially, analysis of ECG waveforms for driver drowsiness detection were performed using wet electrodes attached to the driver.However,it has made it difficult to drive effectively in utilizing these electrodes for driver drowsiness detection. Therefore, it is crucial to eliminate the wires so that these drivers can drive unimpeded. Further research on the ECG compatible components required for drowsiness detection led to the development of dry or non contact sensors, which combats the complications that wet electrodes pose. In this paper, a PS25255 non contact sensor was used to capture the ECG required for drowsiness detection. These sensors are incorporated on the seat back and seat belt with appropriate conductive fabric.

ii. **Filter Stage 1 :** Filter stage 1 is employed at the input terminal after the ECG sensing stage in order to remove 30HZ of electrode contact noise and muscle artifacts. Due to the presence of these noises, the ECG signal is corrupted.Hence, a low pass filter with a cutoff frequency of 33HZ was used in order to remove the noise.

iii. **Amplifier Stage:** In order to improve the strength of the ECG signal for accurate drowsiness detection results, we used the amplifier stage. In this study, the amplifier stage was implemented in two stages. In the first stage of amplification, the input ECG signals are weak; therefore, it was observed that they should be strengthened in order to facilitate the analysis. In the second stage (also called the differential stage), the first stage output is amplified by minimizing the sensor's common potential. Moreover, the process of diminishing the interference depends on the type of differential amplifier employed in the ECG detecting hardware's input stage. We used the unity gain stable OPA 2277 for amplifier stage 1 because of its high common-mode rejection, output free from phase inversion, ease of use and high performance. OPA 177 for amplifier stage 2 because it is unity gain stable and has high performance.

iv. **Filter Stage 2:** Filter stage 2 is employed after the amplifier stage to remove 60HZ of powerline noise, which results in false positives during the analysis. Thus, a twin-T notch filter with a 60Hz cut off frequency was used to eliminate the powerline noise.Additionally, a variable quality factor is achieved with a potentiometer that not only enhances the efficiency of the filter, but also reduces errors. Broadly, this adjustment will give flexibility to the board, and in our design, all the resistors contain 0.05% tolerance to avoid frequency drift from 60Hz. Here, we used OPA 2277 for filter stage 2 because it is unity gain, operates up to 36-V supply rails, ultralow offset voltage,offset voltage drift, and 1-MHz bandwidth.

v. **Driven Right Leg (DRL) circuit:**A Driven Right Leg Circuit (DRL) is added to the bio-amplifiers to decrease the common-mode interference. To clarify, Bio-amplifiers measure very low frequency signals produced by the body, and due to the electromagnetic interference, the body operates as an antenna and picks up the 60HZ power line noise. Furthermore, the DRL circuit has an ECG sensor that detects the ECG signals and eliminates interference.

1.2.2 Hardware Simulation

A drowsiness detection system is viewed as a fundamental system with inputs and outputs. Therefore, in order to run this system in a simulation environment, the environment itself must support inputs and outputs that are typically analogous to those of the real-time signals. Consequently, the motive is to simulate the complete hardware in a SPICE simulator of electronic circuits. In our study, we used LT spice freeware computer software, which implemented a SPICE simulator. The step-by-step procedure for the simulation is shown in figure 7.

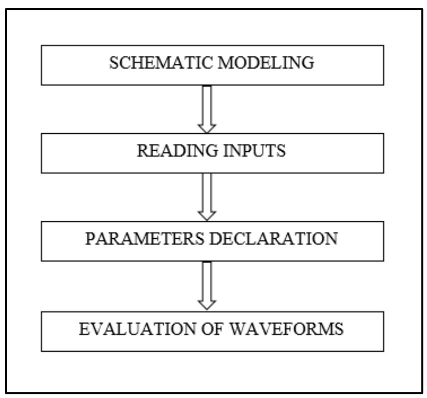

Figure 7. Simulation Steps

In LT Spice, the design model was initiated by a schematic capture. The input ECG samples were collected from different drivers from various driving scenarios, and the values that were obtained were converted into an excel file using MATLAB software. In the subsequent step, this file was converted into a text file and given as the input (it is necessary to note that the transient time and AC analysis are the parameters used for the simulation). The waveforms were generated for each of the design's stages, and they were further analyzed for drowsiness detection.

i. **Input Stage:** In this simulation, the ECG signal with a peak amplitude of 150mV was applied differentially to the system's input. These signals comprises muscle noise and powerline noise. Figure 8 represents the input of the ECG signal applied to the circuit.

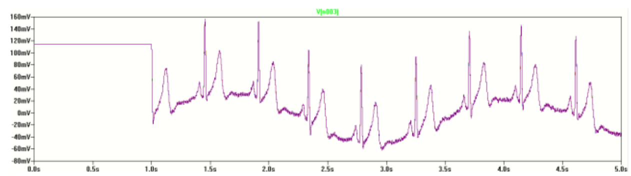

Figure 8. Input signal

ii. **Filter Stage 1:** The applied input passes through two low pass filters with a cutoff frequency of 33HZ and these filters suppress the interference due to muscle artifacts. In this stage, the peak output voltage of the ECG signal is 150mV. Figure 9 represents waveform at filter stage 1.

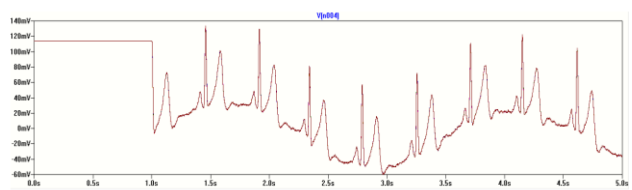

Figure 9. The filtered output from filter stage 1

iii. **Amplifier Stage 1:** After passing through the low pass filter, the input then passes through amplifier stage 1. Principally, these amplifiers improve the strength of the two ECG signals, resulting in a peak output voltage of 1.4V. Figure 10 displays the output at amplifier stage 1.

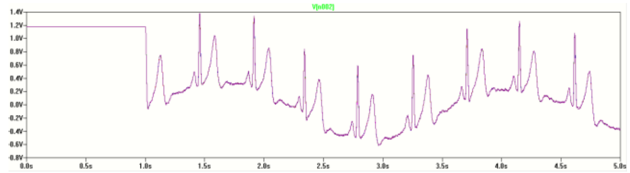

Figure 10. The output obtained from one of the stage 1 amplifier

iv. **Amplifier Stage 2:**Once the ECG signal passed through amplifier stage 1, the signal passes through a differential amplifier in order to minimize the common voltage between the two sensors. The peak output voltage of this stage is 2.8V. Figure 11 represents the output obtained at amplifier stage 2.

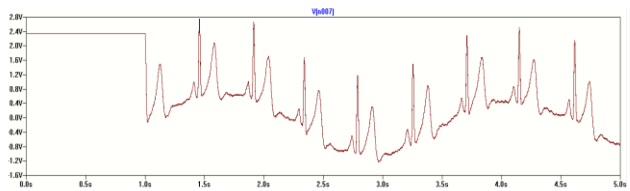

Figure 11. Output after amplifier stage 2

v. **Filter Stage 2:** Due to the power line interference, ECG signals are mixed with a 60HZ power line noise.However, this noise can be removed using an active twin T notch filter with a cutoff frequency of 60HZ. After passing the output of amplifier stage 2 through a notch filter, powerline interferences are removed.Hence, the peak output voltage of this stage is 2.8V. Figure 12 portrays the output after the notch filter stage.

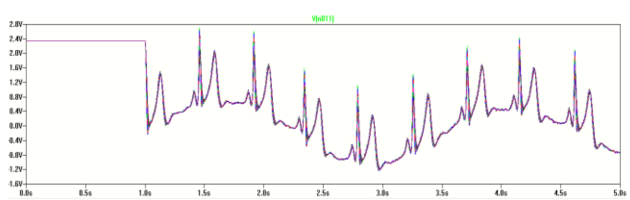

Figure 12. Output after notch filter stage

vi. Output: The output of the proposed circuit was adjusted by using a potentiometer at the output stage. The peak output voltage of this stage is 2.8V for diverse potentiometer values. Figure 13 shows the potentiometer output.

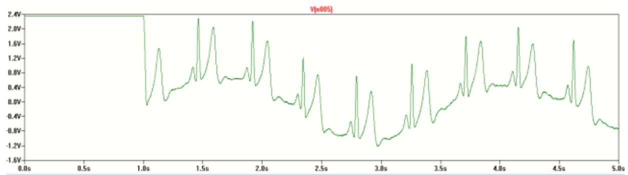

Figure 13. Output for a value of the potentiometer

From figure 13, it is evident that the proposed circuit produces an output ECG signal with peak amplitude of 2.8V, which is sufficient for drowsiness analysis. Table 1 summarizes the output voltages obtained at each stage.

Table 1. Output Voltages

STAGES	**OUTPUT VOLTAGE (V)**
Input stage	150m
Low-pass filter stage	150m
Amplifier stage 1	1.4
Amplifier stage 2	2.8
Notch filter stage	2.8
Output	2.8

We can infer that the input was processed at each stage to obtain an amplified signal without noise as shown in table 1. Additionally, the results suggest that the output produced from the proposed two-stage amplifier and filter is sufficient for detecting driver drowsiness.

For the simulation, we collected data from different drivers for five-hour period including the subject's transition from awake state to asleep state and the data after sampling is then converted to discrete data. An algorithm for detecting HRV from the data and making the decision between the non-drowsy and drowsy states is done using logistic regression. The developed algorithm shows a consistent accuracy over 90% in 20 seconds. This algorithm was developed by the software team as prescribed by Park et al. (2004).

2. Chapter 2. Applying Logistic Regression Technique

2.1 Feature Extraction

Raudys et al. (1991) indicate that HRV frequency domain stores a lot of valuable drowsiness information. Seyd, Ahamed, Jacob, and Joseph (2008) stated that Short Fourier Transform technique is used to slide a window with a variable size in order to extract the R-R time series features in frequency domain, and then FFT transformation technique is applied along the times series input. The result is a two-dimensional (time- frequency) matrix with R-R time series frequency response. According to the Park et al. (2004), the low frequency to high frequency ratio decreases if an individual goes from an awake state to sleep state. Selecting features to achieve the highest machine learning method accuracy is accomplished by trying different features within the same data set. The selected features such as highest peak, standard deviation and peak-to-RMS value are calculated in the LF and HF zones in the frequency domain. The number of features can be raised by adding other statistical features related to HRV. For example, range and skewness of LF and HF in frequency domain have proved to be efficient additions. Sang-Joong et al. (2014) suggested that the features in the time domain such as time variance, peak-to-RMS value and minimum value in 25% to 75% of a selected window in time domain. However, adding more attributes from frequency domain could possibly lower the output accuracy as shown in figure 14. In this research, many different features in both time and frequency domain were selected and tested.

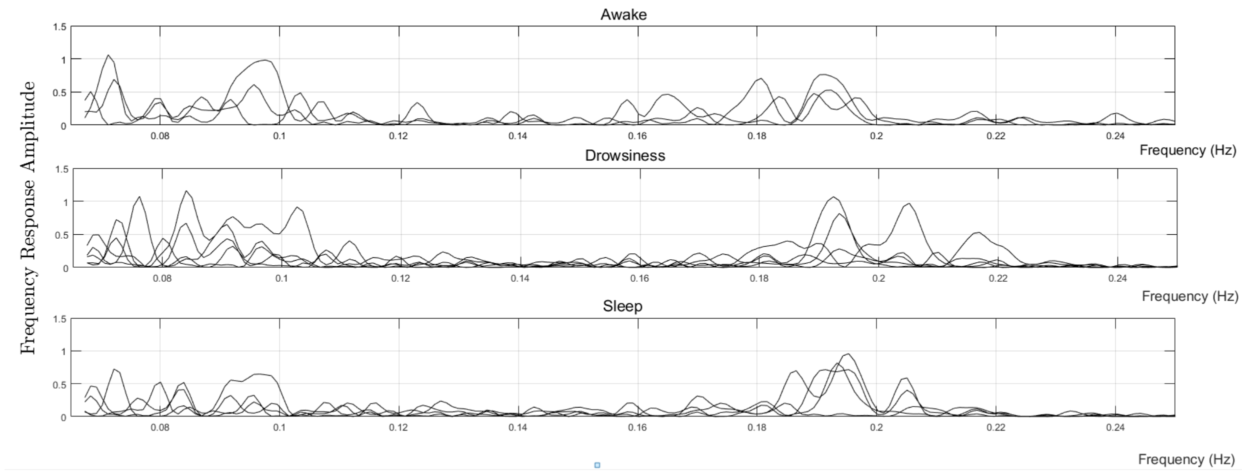

Figure14. High and low frequency domain of peak-to-peak time series

2.2 Algorithm

Since drowsiness has different stages, algorithms relating to likelihood of drowsiness generate a lot of valuable information. We propose both the Naïve Bayes method and the logistic regression method in conjunction because the output value is the probability of drowsiness in both methods. However, the logistic regression technique was ultimately favored and chosen over the Naïve Bayes, which has the best result only if the selected features are independent, and this is not the case with the features with which we are working. Furthermore, logistic regression produces higher accuracy levels than the Naïve Bayes technique does.

2.2.1 Logistic Regression and Prediction of Probability

If we have a set of data in N dimensional space, we define a linear combination of these inputs with arbitrary weight such that

$$s = w^T x$$

And

$$T(s) = \frac{e^s}{1+e^s},$$

Where x is the features, w is the extracted weight, and T is the probability output. Linear output, s, is mapped to a different space which is limited between 0 and 1. The variable S is called the odds ratio, and it is the product of the logarithmic ratio of the probability of drowsiness and the probability of being awake. This parameter is a linear combination of inputs and is written as: S=a+∑b(i)*X where a(Intercept) and b(i) are generated by the algorithm, and X is the input matrix. In this case, learning the exact algorithm is achieved with minimal error with a probability function defined as

$$p(y|x) = \theta(yw^T x)$$

Where y is extreme drowsiness and is assigned a value of +1. Two deep sleep and awake states are being assigned the values +1 and 0. The likelihood of each state is defined by the probability function$p(y|x)$. In cases, where there are multiple data sets, there are n sets, where

$$(x_1, y_1) \ldots . (x_N, y_{\mathrm{N}})$$

Which are independent measurements. Therefore, the likelihood of the total measurement is the multiplication of each individual data set measurement defined as

$$\prod p(\, y_n \,|x_n).$$

In order to maximize the likelihood, the inverse logarithm is differentiated with respect to x(n), and the results evaluates the coefficients and also non-linearity function in the logistic regression method. Linear regression avoids over fitting errors with elimination of large coefficients in linear stage and mapping all outputs to an interval, the number of independent experiments increases the accuracy of the algorithm. Figure 15 summarizes the process of drowsiness detection from the start of recording the driver's ECG data through the data processing, and ending with the determination of the driver's state of drowsiness.

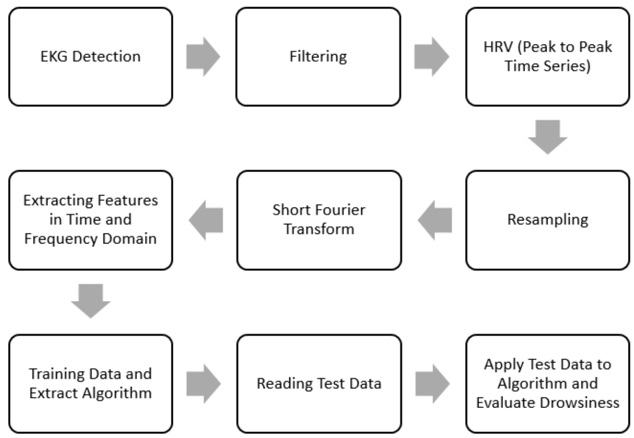

Figure 15. Flow chart of drowsiness detection and algorithm 1 implementation

Algorithm: To detect HRV and for making the decision between drowsy and non-drowsy state

Input: Read ECG Signal (from two different sensors)

Output: The likelihood of drowsiness

// read normalized ECG signal.//

---

1 *ECG_signal = normalized (ECG)*// filtering the ECG signal.//

2 *Filtered_ECG = adaptiveLMS (ECG_signal)*// defining the sampling rate.//

3 *Fs = samplingRate*// calculate time interval between R peak to peak .//

4 *R_to_R = HRV (Fs, Filtered_ECG)*// resampling time series.//

5 *RR = resamplingHRV (R_to_R)*

// extracting frequency domain features

6 *f-features = freqAnalysis (RR)*

/* f-features:

f1 = absolute magnitude square of Short Fourier Transform

f2, f3 = peak magnitude frequency response

f4, f5 = standard deviation

f6 = peak to RMS ratio

f7 = high magnitude frequency response

/* f-features

// extracting time domain features

7 *t-features = timeAnalysis (RR)*

/* t-features:

t1 = average signal in every 64 points

t2 = RMS signal of 64 points window

```
    t3 = standard deviation in every 64 points

   /* t-features

  //creating instances for logistic regression

Instances (:) = createInstance (f-features, t-features)

  //creating data for logistic regression

TotalData(:) =[Instances (:),Label(:)]//splitting data for training and testing.//

Training Data = 90% TotalData

Testing Data = 10% TotalData// calculating intercept and model's coefficient.//

B(:)=logistic_regression(TrainingData(:)). // calculating odds for a new set of data .//

Logit (:)=B(0)+sum(B*Test Data) // Output: Probability of drowsiness. //

Probability(:) =inverse(1+exp(-logit(:)))
```

2.2.2 Experiment

We developed a hardware system with two different setups, one with contact sensors and one with non-contact sensors. In order to detect the ECG signal, remove unwanted noise and also gain the sensor ECG output, we tested two different micro-controllers, Beagle Bone Black and Arduino. We collected ECG signal data using each device for a five-hour period that included the subject's transition from being awake to being asleep. That was converted to discrete data. The Arduino's sampling frequency of 10 kHz was much lower than that of the Beagle Bone

Black, 200 MHz. Therefore, the data took up much less space and speed up the system performance. The program was developed in MATLAB 2015.

Table 2. Time to detect drowsiness using machine learning system

Time	**Machine Learning**	**Subj. 1**	**Subj. 2**	**Subj. 3**
Delay to detect(s)	Yes	28	24	30
Delay to detect (s)	No	145	138	132

Table3. Coefficient of logistic regression analysis

Subject	**Subject 1**	**Subject 2**	**Subject 3**
Window_mean	1868	1734	1441
Window_std	619.4	589.2	673
Window_rms	-1496.5	-1567.2	-1460
Peak High Freq	112	132	150.5
Peak Low Freq	6.4	7.2	14.29
Std_HF	-80.1	-74.4	-82.6
Std_LF	-59.4	-68.2	-10.36
Peak2rmsHf	-4.5	-5	-7.16
Peak2rmsLF	-3.4	-3.2	-9.85
Intercept	-272.0	-293	-199.83

The ECG data was collected from three different individuals of different ages and genders, and the system was tested with two different methods, logistic regression and Naïve Bayes. The result turns out to have higher accuracy in case of logistic regression. Since Naïve Bayes method relies on independent features and our features are dependent in frequency domain. Therefore, the result confirms the lower accuracy of Naïve Bayes output. The accuracy of output for logistic regression test is over 92%. However, Naïve Bayes accuracy never exceeds 85% accuracy. Moreover, the minimum time of drowsiness detection in our set-up was 12-24 seconds. For every new person, the system collects few sets of ECG data while the subject is asleep and train the data to create a unique algorithm for each person. In the testing mode, the system detects between 12 and 24 seconds of ECG signal and immediately publishes its prediction as the probability of drowsiness, as illustrated in figure 16. The experiments result for three people shows a consistent accuracy over 90% in less than 24 seconds. Our developed algorithm provides steady results over time detecting drowsiness (approximately 20 seconds detection). The algorithm and feature extraction are listed in tables 2 and 3.

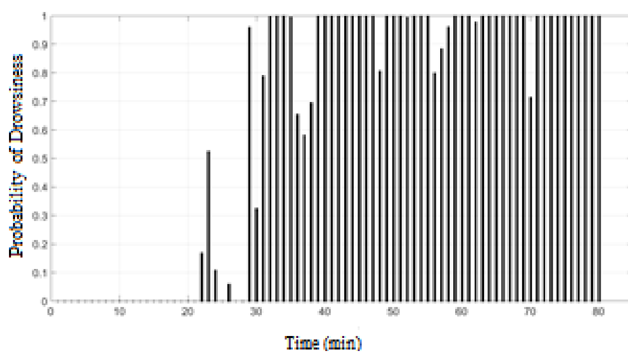

Figure16. Probability of drowsiness

3. Chapter 3. Wavelet and Ensemble Logistic Regression

Wavelet techniques initially introduced by Crosier, Esteban and Galand in effort to decompose discrete time signals in 1976. In this method, unlike short Fourier transform, a window slide along a signal with different sizes. Babaeian, Francis, Dajani and Mozumdar (2019) stated that wavelet technique is useful to decompose non-stationary signals. Due to nature of bio-medical signals, such as EKG, these signals could vary over time for various reasons such as physiological states. Therefore, EKG is a non-stationary signal and applying short Fourier transform is not going to retrieve the accurate frequency response in different times (The frequency features changes over time). A time-frequency domain analysis measures the frequency responses in different time variation. In short Fourier transform, the time resolution is constant. Therefore, resizing windows in different timing results is equivalent to passing signals through low to high pass filter in different time domain. Due to uncertainty principal, the exact time and frequency can never be calculated. Therefore, when we dilate the window, the time interval picks up lower frequency interval with higher resolution. In our analysis, we used 7 level dilatations (2 to 2^7 levels). The number of levels is named as the number of voices per octave and the highest level of dilatation is named the total number of octaves. In this case, the number of voices per octave is 16 and the total number of voices is7. The 2,4 and 8 scales are corresponding to HF level and 16 and 32 are corresponding to LF level. In our analysis, we slide the window ¼ of the window size in each step to avoid overflowing the memory and also to have each portion of HRV to share the wavelet analysis 4 times with preceding HRV data. Finally, over each time slot, we average each statistic over different resolutions in order to extract the updated statistics.

3.1 Challenges

Noise affect negatively the system's output accuracy and it is very difficult to eradicate the noise while using the non-contact sensors. According to Lee et al. (2019), the disruptive mobile environment of a car absorbs noise interference and errors can occur if any relevant data is lost in the transition in removing the background noise in post-processing. Furthermore, in the transition between alertness and drowsiness, the system needs to obtain ECG data in the least amount of time. Therefore, it is crucial that the amount of data collected is extremely low and that the machine learning system has a low number of instances from which it can be trained. However, the main challenge is developing the algorithm with the highest possible accuracy by using the existing collected data. Li and Chung (2013) stated that the environmental noise varies temporally and affects the drowsiness detection efficiency. Individual physiological and psychological states vary, which in turn affect the accuracy of the algorithm. Therefore, the proposed model must be designed to continuously update itself using the incoming ECG data stream, and since ECG is a non-stationary signal, a new technique by applying Wavelet transform to extract frequency features in the HRV's LF and HF areas. This helps to minimize the detection time and an increase in the system's accuracy. Then, Khazaee et al. (2010) suggested a new method to extract features in frequency domains using fixed sized windows and selecting the support vector machine (SVM) classifier for the generalization process. It depicted an SVM classifier with a Gaussian rule-based fuzzy (GRBF) kernel, which achieved over 95% accuracy, if the noise interference from the training and test data is correlated. Otherwise, the system's accuracy may plummet to 70 percent. Jen et al. (2008) proposed utilizing the neural network classifier to extricate optimized features in order to achieve a higher accuracy. However, the system required collecting ECG data over a long-term period. Zhao et al. (2005) tested a method for feature extraction by using wavelet transform and applying SVM, but while the proposed approach attained an overall 98% accuracy, the method's complexity had a slow processing time. They implemented a new algorithm to process HRV data with wavelet transform, and because of the wavelet transform's effectiveness; they achieved 95% accuracy in a one-minute time frame. From a detailed perspective, it is clear that a support vector machine was applied to obtain this accuracy. By applying a low complex Logistic Regression (LR) classifier, Babaeian et al.(2016) formulated a new approach by sliding a fixed window along the signal to extract features. They further aim to minimize time detection and increase accuracy with the new techniques described in methods. The new algorithm incorporates ensemble logistic regression (ELR) with the regularization technique to overcome the over fitting problem and also employs wavelet technique over short-time Fourier transform (STFT) to adopt the non-stationary nature of ECG. Due to the low complexity of the LR technique, the algorithm's speed increases. However, if the number of features exceeds the number of instances, the LR algorithm is more likely to be over-fitted. In that case, the algorithm could have a low accuracy with different ECG data.

3.2Contribution

Earlier research methods using only the LF/HF ratio indicator required data collection periods of two or more minutes resulting in a drowsiness assessment accuracy that approximated 60 to 70 percent. Previously, we had developed a method to minimize the detection time and increase accuracy by applying a Logistic Regression (LR) method with a STFT technique. Yet, while recent studies have lessened the error to less than ten percent, they did not account for the data's size. From a thorough analysis, there seems to be a trade-off between reducing detection time and increasing accuracy in order to alleviate this issue. This study proposes a new algorithm to accomplish reduced detection time and an increase in accuracy. On an average, the human heartbeats around 60-100 beats per minute. Therefore, in order to make an accurate detection, previous methods required substantial HRV data implying that the system must wait substantially to detect drowsiness. This time frame may be extensive enough to cause a fatal accident. Therefore, we have exercised an ensemble logistic regression technique to detect a driver's drowsiness in less than a minute. Xu and Frank (2004) stated that ensemble logistic regression technique handles a low number of instances despite the exchange between minimizing detection time and increasing the system's accuracy. In this study, we describe our recently developed ELR algorithm, which uses a small amount of raw ECG data generated in less than one minute of monitoring while retaining over 90% accuracy. Contrasting traditional methods that used a LF/HF indicator, this machine learning technique employs a vast number of features to increase accuracy, and as the results indicate that accuracy improves when applying the machine learning technique over the LF/HF indicator. Furthermore, electrocardiogram signals are non-stationary and vary in response to distinctive environmental effects as well as personal, psychological, and physiological states. Hence, it is more effective to replace STFT and Fast Fourier Transform (FFT) with WT (Wavelet Transform), a more potent tool for extracting features in a frequency domain. By substituting the WT for STFT techniques, we are addressing the non-stationary nature of the ECG signal. Contrarily, reducing the detection time leads to a small data set to train the algorithm. As a result, the number of features exceeds the number of instances, resulting in an over fitting problem when using the LR technique. Therefore, to overcome the issue to work with a smaller ECG sample, the solution is to use ELR instead of an LR method. Because the sample data set collected during the algorithm training session is small, deterministic noise survives the processing and is still present in the final algorithm. Therefore, in order to solve this problem and improve accuracy, we have applied regularization in addition to a distinctive technique for feature selection to the algorithm. Environmental conditions as well as the driver's psychological and physiological states are continuously generating new and different stochastic noise. Therefore, the algorithm training sessions were programmed to continue indefinitely allowing them to adapt to these alterations. This approach forms a complete, real-time drowsiness detection system, consisting of both the hardware capable of detecting the heart's electrical signals and the software that is adept in predicting drowsiness. In light of these observations, it is necessary to note that while maintaining over 90% accuracy, this drowsiness-detection method is able to use its collected data in less than a minute. This is due to the fact that a drowsy driver could find him or herself in a hazardous situation before the required data could have been collected and processed. Moreover, it would be trying to collect the required data for training purposes in a short period of time while including both alert and drowsiness states. Employing the machine learning technique and the ensemble-based system implements the algorithm to minimize detection time with the highest possible accuracy. Moreover, applying ELR is an innovative technique that allows for us to employ a machine learning algorithm to detect driver drowsiness. Our proposed approach, applying this new technique:

- Lowers the detection time by developing a unique algorithm in which utilizing a short time frame of an ECG signal is sufficient enough to detect drowsiness.
- Increases accuracy by improving the feature selection technique.
- Detects the level of subject drowsiness in various stages (a technique known as multi-stage detection).

- Lowers the algorithm-training period while collecting the driver's ECG. This reduces the drowsiness detection period.
- Accounts for the lower sample size by adding regularization to the final algorithm; this is done to balance the effect of deterministic noise.
- Continuously updates the algorithm to account for human physiological and psychological states as well as environmentally induced stochastic noise.

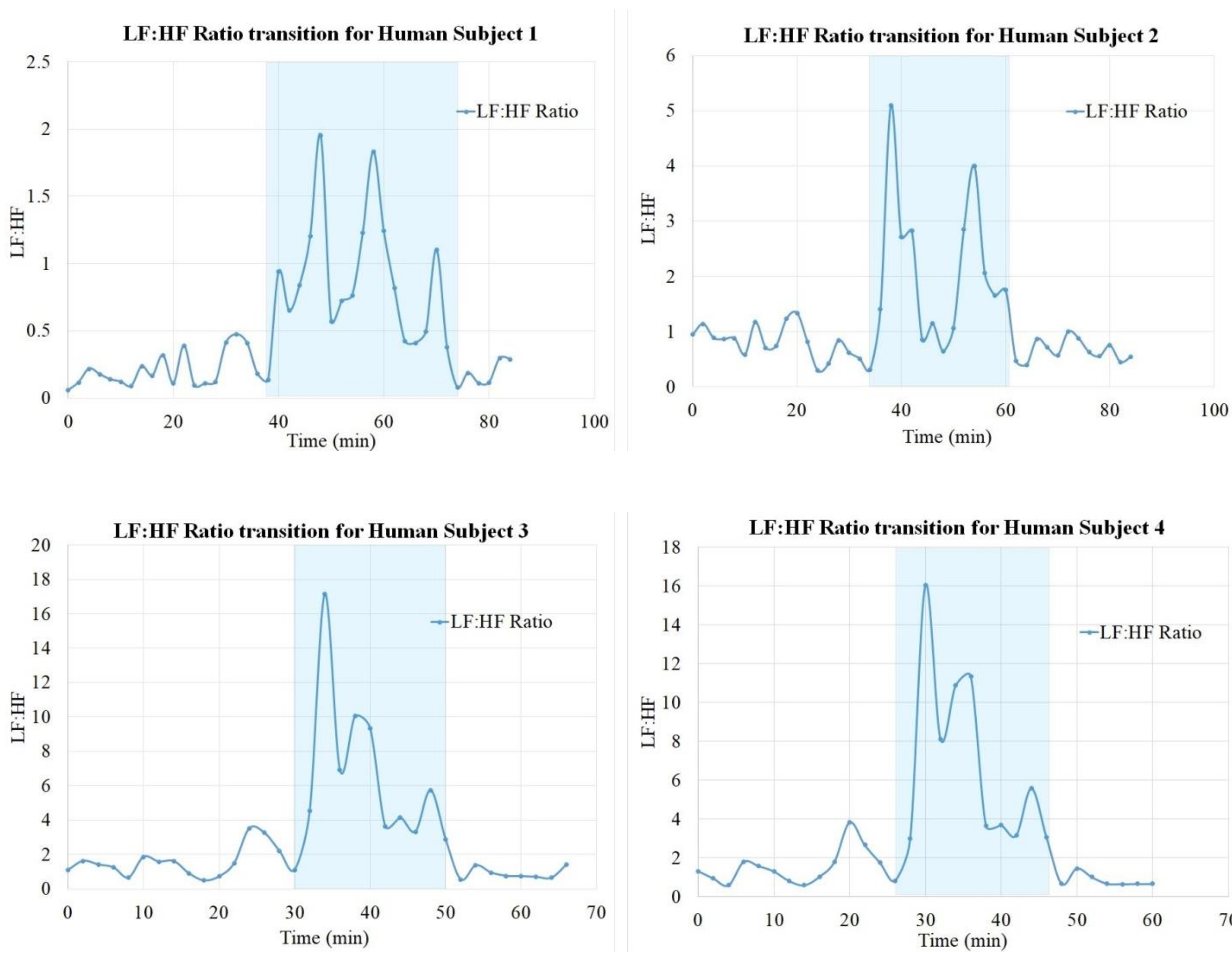

Figure 17. Graphs illustrate the changes in the LF to HF ratio

3.3 Methods

Our proposed algorithm aims to improve accuracy by eliminating the complications that result from collecting a small amount of data during the training period. Specifically, it implements an analysis based on wavelet techniques to extract features from a non-stationary ECG signal resulting in a higher resolution in time and frequency domains. In the beginning of the process, the hardware sensors detect the driver's ECG signal and then, the hardware filter removes power line noise. Afterwards, a software adaptive filter eliminates most of the noise-causing distortion in the ECG signal. Secondly, we extract the HRV from the ECG. To generate a time series in equal intervals and apply a frequency transform process, HRV is resampled to generate a time series with equidistant intervals in the time domain, and the resampled HRV time series is generated using an interpolation technique. Afterward, a Continuous Wavelet Transform (CWT) is applied to extract frequency domain features in high and low frequency regions. In this algorithm, we used fifty-two features from time and frequency domains based on their impact in the LF and HF areas of the collected HRV data (e.g. peak, energy, and range). This unique algorithm selects the most appropriate features to classify the training ECG data set with binary labels (alert and drowsy). Subsequently, the selected features are used in the logistic regression function in order to produce a Machine Learning (ML) model minimalizing the output error.

The sleep cycle consists of three major periods: awake, non-rapid eye movement (NREM), and rapid eye movement (REM). NREM represents light sleep, just after falling asleep. REM consists of deep sleep and vivid dreaming. The drowsiness stage is the transition period between awake and NREM and vice versa. Usually, the drowsiness stage is detected when a subject's body movement slows significantly. In this experiment, ECG data was collected from thirty healthy human subjects varying from ages twenty to sixty years old, and the measurements were taken from a three-lead ECG measuring device. Each subject was taught to apply the ECG leads and collected their own data, unsupervised in their normal place of sleep, until they naturally woke up, completing the awake-sleep-awake cycle. Throughout the data collection period, the subjects were undisturbed and resting, and during this period, the subjects began alert, gradually fell asleep, and then woke up after a period of time. Through video-recorded observation, the times at which the subjects fell asleep and awoken were noted. Each subject produced between 6,000 and 10,000 data points. Each point was time stamped and designated alert or not-alert, as determined by both Fitbit data and a video recording of the subject. At the time of this experiment, Fitbit was the most advanced and well-tested commercially-available device for tracking sleep states. The transition period from awake to sleep or from sleep to awake, also called the drowsy state, is the area of focus. This transition period lasts from a few seconds to a few minutes. The commercial activity tracker called the Fitbit bracelet has the ability to record and monitor sleeps stages. This ability is based on tracking of body movement and heart rate variation. Additionally, a camera detected the subject's eyelid movements, which were used to evaluate the transition periods. As the eyelid movement stops, the subject is labeled asleep. This data label is confirmed by the Fitbit's assessment and if they do not match, the data set was not used, and the subject was instructed to recollect data. This information was utilized in order to label the data to train the ML model. Therefore, comprehensive awake-asleep-awake ECG data was obtained for analysis, and according to our observations, the subjects' maximum LF to maximum HF increased consistently as they fell asleep. This confirms the accuracy of the collected data as demonstrated in previous research. As indicated in our calculations, figure 17 demonstrates the change in the ratio of LF to HF for four subjects, and the collected data shows three dissimilar periods: awake, sleep, and awake. In order to develop the algorithm, the transition times between each state (these times are less than a minute) were sampled and studied. Results have indicated that the change in drowsiness states in each subject occurs in a short period of time. For example, the transition period for Human Subject 1 occurs precisely at thirty-eight and seventy-two minutes after the sampling process has begun (Human Subject 1, Figure 17).

The ratio of LF to HF is often used to evaluate an individual's drowsiness level. However, it requires at least two minutes of data collection before the first sign of drowsiness can be detected. This delay in temporality is extensive enough to counter the primary purpose of this research and to quickly detect drowsiness and intervene before the vehicle and the driver are deleteriously affected. Therefore, ELR was selected for this method not only to reduce the training period time needed for drowsiness detection, but also to increase the system's accuracy. In our previous research, the detection time was minimized to less than one minute by sliding a fixed window along the signal and then applying STFT. However, this method's accuracy diminished as soon as the size of the window decreased, which is done for faster detection. Therefore, this method improved the accuracy of the results by applying a variable window size with wavelet techniques. In this method, we resized the time window to at least two HRV samples, extracted features in the time and frequency domains, and dilated the window until the desired accuracy threshold was achieved. In this algorithm, we averaged the selected features over different window sizes, and we observed the varying results in the differing window proportions, the WT technique window resizing reduced the early detection period as low as twenty seconds, but with the same accuracy as a thirty-five second fixed window in STFT technique.

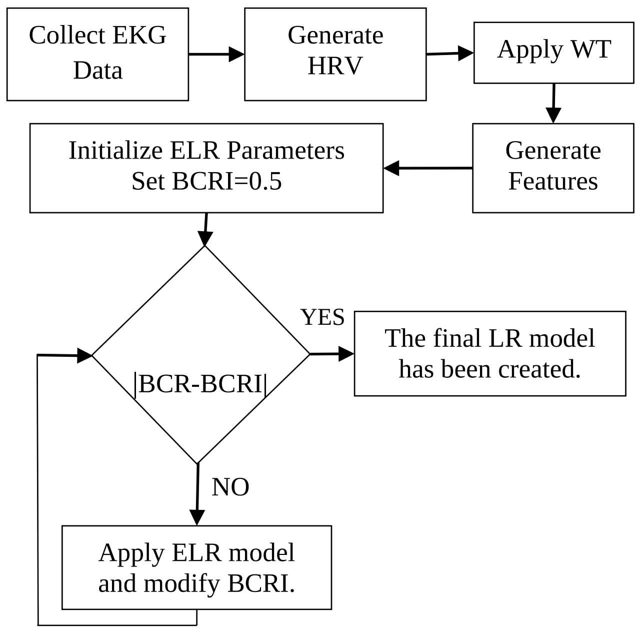

Figure 18. Flow chart demonstrates the algorithm

In the algorithm's initial steps, lines 1 through 5, data acquisition, data normalization, and filtering were accomplished in addition to the R peak detection and resampling of the HRV data; this was followed by the application of Wavelet transform (denoted by line 6). Once the features are generated, it is imperative to form a probability vector for feature selection which is followed by establishing the logistic regression model. To overcome certain drawbacks such as the over fitting of logistic regression due to a small dataset, regularization was applied. Based on the t test technique, the initial probability vector was formed and to achieve efficient feature selection, a loop was implemented, which sorted the features based on decreasing probability values of the p-vector. Then, the data was divided into two sets of data labeled Training and Testing. These data sets were used to extract the BCR values shown in steps 14 through 16—thereby determining the specificity and sensitivity, and after these steps, we updated the p-vector with the new BCR value in lines 17 and 18. The loop ends when the discrepancy in BCR's consecutive values converges. Lastly, the LR model is provided with ECG signals to continuously compute drowsiness. Figure 18 shows the order of these processes.

3.3.1 Wavelet Transform

Historically, Wavelet Transform (WT) techniques have only been around for a few decades. This method, contrary to STFT, uses a window of variable sizes that slides along the signal and filters it out. After defining a window (as represented by φ) and if the wavelet transforms of $g(t)$ are obtained, the continuous WT is defined as follows:

$$g(a,b) = a^{r} \int_{-\infty}^{\infty} g(t)\, \varphi\left(\frac{t-b}{a}\right) dt,$$

The variable φ, also known as a mother wavelet, forms the orthogonal basis. There have been numerous known windows for WT, one being defined as the Morlet mother wavelet. The Morlet window is described as:

$$\varphi(t) = e^{iat}\left(e^{-\frac{t^2}{2\sigma}}\right),$$

Where t , a , σ, and r represent time, the scaling factor, mother wavelet variance, and the normalization factor, respectively. While variable a connotes frequency and b signifies time resolution.

Due to the nature of bio-medical signals such as ECG, these signals can vary over time for various reasons, including but not limited to changes in physiological states. Li, Zheng and Tai (1998) stated that ECG is a non-stationary signal, applying STFT will not retrieve the accurate frequency response for different time intervals, and since frequency features are not static, a time-frequency domain analysis, such as WT, measures the frequency response of the signals in different time intervals. In contrast to STFT, the time resolution achieved through the WT varies. So, resizing windows during the translation parallels passing signal through low to high filter in a different time domain. Due to the Heisenberg uncertainty principle, one cannot know the exact frequency instantly, but we can discern what band of frequencies exists in specific intervals of time. Therefore, when we dilate the window in WT, the time interval picks up lower frequencies with higher resolutions while compressing the window reveals higher frequencies with a higher resolution. In this experiment, the window size was halved seven times to achieve a higher resolution. Furthermore, the number of levels were named according to the number of voices per octave (sixteen per octave), and the maximum level of dilation was labeled the aggregate number of octaves (totaling seven). The number of voices per octave represents the number of separations between the minimum and maximum scales. The two, four, and eight scales corresponded to the HF level, while sixteen and thirty-two corresponded to the LF level. In our analysis, we slid the window up by one-fourth in each step both to avoid memory overflow and to have each portion of HRV share the wavelet analysis four times with preceding HRV data. As shown in figure 19, the scale-transition plot demonstrates the LF/HF ratio changes in subject 1's alert, drowsy, and sleep stages. Furthermore, the dilation scale proportionally represents the inverse of frequency variations, and the transition signifies the time changes that occur while the window slides along the signal. Lastly, over each time slot, we averaged each statistic over different resolutions to obtain the final statistical results.

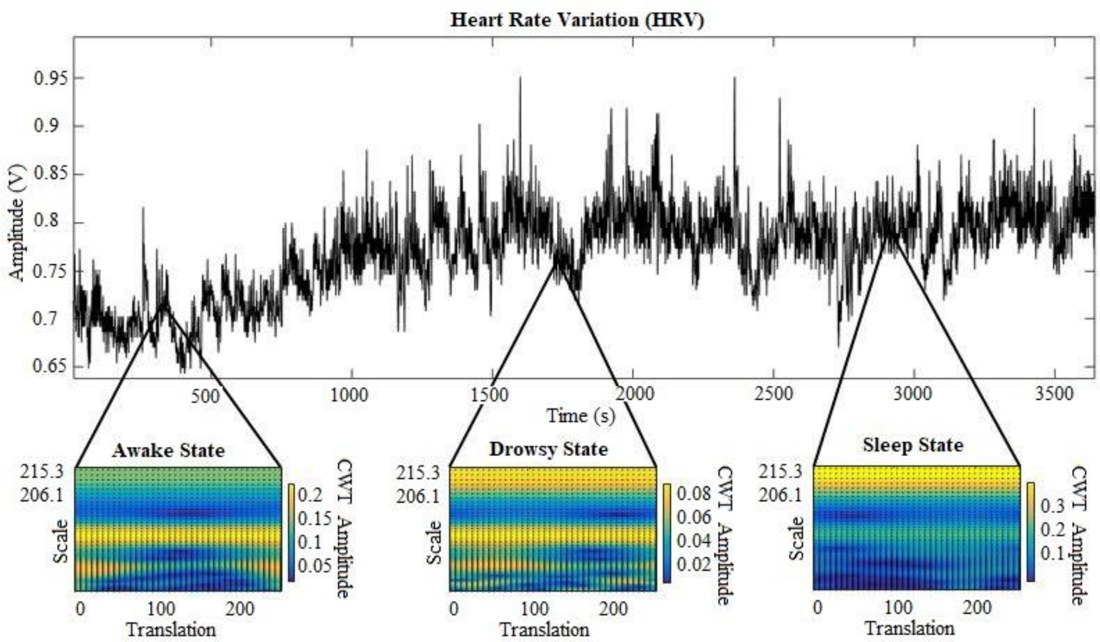

Figure 19. Graph illustrates the HRV of a subject along with the CWT graphs

3.3.2Logistic Regression

Logistic regression is a linear machine learning technique used to find the best hypothesis. In regression methods, a group of total instances is formed in conjunction with a binary label for each data set. This is a supervised machine learning method in which the data training set labels are known. In the proceeding step, a new data set based on a non-linear wavelet and STFT transfers the HRV data to a new space with the same label.

$$(x_1, y_1 \dots x_n, y_n) \rightarrow (z_1, y_1, \dots, z_n, y_n)$$

Where x represent instances, y denotes features, and z is the output label.

By defining

$$L(y) = sign(w^t y)$$

And

$$LT(y) = \frac{1}{\left(1 + e^{\left(-L(y)\right)}\right)}$$

y is the selected feature. The output result is a probability of occurrences consisting of two states: alert and drowsy. We also defined the above L function as the loss function, which is linear in w. To clarify; the w vector is calculated based on the steepest gradient descent method used to minimize the L function with respect to w.

3.3.3 Regularization

Principally, regularization is a common method meant to avoid data over-fitting, and in this study, deterministic noise has a significant effect on the accuracy of the system's final output. The output error value represents the dichotomy between the predicted result and the actual output and consists of two terms: variance and bias. While variance is a measurement of how scattered the functions are within the set of possible algorithms in a single machine learning method, bias is how distant this function set is from the ideal function. Malik (2004) stated that regularization techniques aim to minimize variance, which equates removing deterministic noise. This was done by adding a constraint to our model, and the limitation was due to the existence of a low number of samples. As demonstrated in the algorithm, a common method for minimizing variance is the L2-LR technique, and this method adds a constraint countering the larger weights against the small changes in instance input samples. In LR analysis, adding an extra term to the regression loss function generates a new loss function. Therefore, it must be minimalized with respect to w so that $Loss(w) = 1 + e^{(w^T X)} + \lambda \sum_{i=1}^{n} w$.

Applying the descendent gradient method gives

$$w = (X^t X + \lambda I)^{-1} X^t Y$$

Where w represent the weight of features in the ELR function. The proper amount of λ delivers the best regularization. Moreover, while the variable λ is an arbitrary parameter, increasing it helps resolve the over-fitting issue caused by a small amount of data. However, the algorithm loses higher degree terms as λ increases, making the under-fitting too severe for regularization to effectively avoid over-fitting.

3.3.4 Ensemble Logistic Regression Method

In integrating a machine learning technique, LR is a dominant algorithm that classifies binary output with a likelihood of occurrence. But, while the LR method has a high accuracy and simplicity function, it tends to over-fit and implements an erroneous algorithm when the number of features exceeds the number of instances. ELR, a new technique developed by Zakharov and Dupont (2011), was used to select the most superlative features in order to overcome overfitting. Since, feature selection contributes to the LR's output accuracy, the number of features must be close to the number of learning instances, and when the number of training samples (represented by the variable n) is less than the number of features (p). Some of the features have dominant weights over other features. Therefore, a small alteration will have a significant impact on the output. However, it is crucial to note that this technique does impose a penalty on the output while still allowing for over-fitting, which commonly known as regularization reduces the effect of large feature weights. If there are p features of $x(i)$ where

$$0 < i < p$$

Then, we define the LR classifier as

$$Prob(y|x) = \frac{1}{1 + e^{\left(-y(w^T X)\right)}}$$

Where$w$ (i) are weights extracted by the LR classifier. Therefore, the loss function is defined as

$$Loss(w) = 1 + e^{(w^T X)}$$

If we add the penalty factor to the loss function, then we have

$$Loss(w) = 1 + e^{(w^T X)} + \lambda \sum_{i=1}^{n} w$$

Where λ is an arbitrary parameter. The second additional term $(\lambda \sum_{i=1}^{n} w)$balances the feature's effects with a higher$w$ (i). The optimum λ is normally tested using a few arbitrary numbers however λ is set to a fixed number in the entire iteration. The algorithm implements a t-test technique. The L2- regularization usually provides unique answers to the desired model and is robust to outliers. However, L1- regularization does not have the same stability, but has tendency to lower the weights to zero. Therefore, it eliminates the features which have small weight and in other words, it automatically applies a feature selection method. This method employs L2-regularization method and adding a new algorithm to apply feature selection. Therefore, this technique has L1- method sparsity along with output uniqueness and reducing outlier's L2-method at the same time. The algorithm constraints the model to the reduced number of features just same as the number of samples.

Where μ is the average and σ is the variance of features for each label$(+, -)$.

$$t_j = \frac{\mu_{j+} - \mu_{j-}}{\sqrt{\sigma_{j+}^2 / m_+ + \sigma_{j-}^2 / m_-}}$$

Where μ_{j+} is the mean expression value of the feature j for m_+ positively labeled examples, μ_{j-} signifies the mean expression value of the feature j for m_- negatively labeled examples, and σ_{j+} and σ_{j-} are the associated standard deviations. The score vector t over the p features is normalized to produce a valid probability distribution vector; hence, the result is normalized and called the probability vector. According to the highest probabilities in the Prob vector, the algorithm selects the features of the selected subset of instances for each iteration. Then, the time series HRV is divided into two groups: the training (80% of the sample subset) and test (20% of the sample subset) instance groups. W's subsets are generated according to l2-regularization LR, and its complexity grows linearly with the number of features. Therefore, an ascendant gradient technique converges to the most updated $w's$, and lastly, a quality term BCR (initially set to 0.5) is calculated and updated upon validation. We applied the training set to calculate the LR model output as well as the model's sensitivity and specificity.

Subsequently, the balanced classification rate (BCR) is defined as:

$$BCR = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

Where TP and TN are the numbers of correct output readings among the sample subsets of P and N respectively. The probability vector is updated with the new BCR, and, it converges after little iteration. Finally, after convergence, the loop ends, and the final probability vector selects features. In this algorithm, the 20% validation set generates specificity and sensitivity to update BCR and then probability vector. This 80- 20 split is not for the final result since this 80-20 split repeated over many additional iterations until BCR converges. Many numbers of iterations are using the same features training and validation sets over and over. Therefore, the same set gets to mixed up and split to 80-20 groups multiple times. By choosing lambda a fixed number in each iteration, the sparsity is satisfied by choosing the number of selected features is exactly equal to the number of samples in each iteration.

3.4 Results

To minimize environmental noise in this experiment, ECG samples were first collected and filtered through a notch and adaptive filter. Secondly, the HRV time series was extracted and resampled to generate a new time series and lastly, STFT and wavelet analysis was used to extract the features. Broadly, we collected over 6,000 HRV samples for each of the thirty subjects in both alert and drowsy states, which included 11 females and 14 males ranging from ages 20 to 60. Additionally, we extracted the following twenty-four features from each of the LF and HF regions.

Table 4. Features from LF and HF regions

Frequency Domain	**Time Domain**
Maximum peak	maximum
Range	variance
Standard deviation	mean
Peak to rms ratio	standard deviation
Mean	
Skewness	
Kurtosis	
Energy	
Peak to amplitude of 0% range	
Median interquartile range	
Median	
Edit distance on real signal	
Maximum to minimum distance	
Seqperiod	
Minimum	
Energy of right side of maximum	
Energy of left side of maximum	
The second largest element	
The distance between the maximum and the second maximum	
The rssq in frequency domain	

We labeled the HRV data with “drowsy” or “alert” with respect to the subjects’ body and eyelid movements.

In the subsequent step, epochs were formed spanning from 2 to 128 HRV data points in each group. In the Wavelet analysis, the features were averaged holistically over the different window sizes and each epoch. We selected fifty-two features according to the t-test technique and applied the LR algorithm with l2 correction LR function (LASSO) analysis to eighteen to twenty-four instances. Then, we split the set of instances to two groups with the ratio of 80% and 20%; the 80% set is the training set, which has 18 instances. Conclusively, we calculated the quality function by applying the sensitivity ratio and the BCR to the remaining six instances. The BCR was constantly updated with twenty-four new features selected by the algorithm's updated output probabilities according to the algorithm.

When applying wavelet, regularization, and ensemble methods, the results indicate that the output accuracy improves between 90% and 95% compared to those obtained by STFT and LR methods in our previous research. Moreover, the detection time drops to as low as twenty seconds, granting a 20% improvement over the LR method. In table 6, we illustrated the output accuracy for thirty subjects in order to compare using the wavelet method with the applied methods that were presented in this study. The accuracy was measured by the ratio of total true labels in the test data set over the total of true and false responses by the model for the same test data set. Table 4 compares the minimum detection time for the thirty subjects. To do this and measure accuracy, we simply lowered the window's size to as low as two HRV samples while the detection time window lowered. Also, as the accuracy becomes less than 85%, we then measured the detection time, but due to the frequency band restriction in the HF area, a detection time using less than eighteen samples results in inaccurate measurements. Hence, lose the information in the frequency domain.

In table 5, drowsiness detection time drops to less than thirty seconds while maintaining accuracy above 85%. This decreases the time detection by at least 20% when compared to that of STFT. This in turn, demonstrates the advantage of WT rather than STFT. Furthermore, the results in table 6 confirm the improvements in the system's error detection by more than 5% in contrast to those from STFT. In all thirty subjects, accuracy increased to 90% and results have shown that WT combined with ELR feature selection creates a reliable technique for drowsiness detection.

Table 5. Comparison based on the time required for detection

	Age	Traditional LF/HF, time (sec)	STFT & LR, time (sec)	WT & ELR, time (sec)
Male	20-40	134-144	24-35	18-24
	41-60	136-148	27-33	21-24
Female	20-40	126-140	24-31	18-26
	41-60	128-146	26-36	20-24

Table 6. Comparison based on the accuracy of detection

	Age	Traditional LF/HF (%)	STFT & LR (%)	WT & ELR (%)
Male	20-40	63.1-70.9	86.0-87.7	91.3-96.8
	41-60	67.0-70.3	84.3-86.4	91.2-96.8
Female	20-40	64.3-68.2	84.7-86.1	91.2-96.8
	41-60	66.0-68.5	84.7-86.8	94.5-96.0

Algorithm for drowsiness detection

Input: The raw ECG Signal (ECGdata, Sampling_Frequency).

Output: Logistic regression function LRmodel.

1.*Normalized_ECG = ApplyNormalization(*ECG_{data}*)*

//Normalizing the raw ECG signal. //

2.*Filtered_ECG = AdaptiveFilter(Normalized_ECG)*//Filtering the ECG signal to remove noise using LMS Adaptive filter.//

3. *RtoR = R_PeakToPeak(Filtered_ECG)*//Computing the R peak-to-peak time series

4. *HRV = Resampling(RtoR, Sampling_Frequency)*//Resampling the R to R time series at a sampling rate of Sampling_Frequency. //

5. $Instances_{[m:int(n/m)]}$ = Epoch($HRV_{[1:n]}$) //Grouping HRV data to apply wavelet analysis; m is number of HRV instances. //

6. [F_1, F_2, F_3, …, F_n] = WaveletAnalysis(Instances) //Extracting frequency based features from HRV using continuous wavelet analysis; F_1 to F_n are frequency domain features. //

7. [p_F_1, p_F_2, …, p_F_n] = Ttest (F_1, F_2, F_3, …, F_n) //Calculating initial probability for feature selection. //

8. Initialize BCR = X //X is any value between 0 and 1.//

9.**do**

10. BCRI = BCR

11. *Sort_F_based_P*(F_1, …, F_n,p_F_1, …, p_F_n) //Sorting features in decreasing order of probability in p_F. //

12.*Training_data = t% of m instances*

//Splitting the original data set into a Training data set. //

13.*Testing_data* = (100-t)% of m instances //and a Testing data set

14. $[W_0, W_1, \dots, W_m]$ = *l2_Logistic Regression* $([F_1, F_2, F_3, \dots, F_m]$,*training_data)*

//Computing the coefficients of the Logistic Regression model using the training data set. //

15. *[TP, TN, P, N] = Logit* $([F_1, F_2, F_3, \dots, Fm]$, *Testing data*, $[W_0, W_1, \dots, W_m]$ *)*

//Evaluating Specificity and Sensitivity. //

16. $BCR = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$//Averaging Specificity and Sensitivity. //

17. *Quality = log(1+BCR-BCRI)*//Improvement factor to update probability. //

18. $p\_F_{[1:m]}$ = Normalized ($p\_F_{[1:m]}$ + Quality * $W^{2\text{sign(Quality)}}$) //updated probability vector. //

19.**while** (|BCR-BCRI|>>0)//Quitting the loop only when BCR converges. //

20.*LRmodel = Logit* //Final logistic regression model. //

4. Chapter 4. KNN and Support Vector Machine

Support vector machine (SVM) is a technique used to generalize regression methods. It was developed in 1992 by Vapnik and his teammates at AT&T Bell Laboratories. Using this technique, a group of observable instances is split into a training set and a test set. The ultimate goal to find the best hyper-plane function to classify the training set (x_i, y_i). If we define this separator function as $f(x) = (w, x) + b$, then there are many different classifiers that separate the two-binary set of groups. However, with the help of SVM technique, we are choosing these parameters to maximize the distance of this function and the nearest points from both binary groups. In this method, if $x_i : i \in \{1: m\}$ and $x'_i : \in \{1: n\}$ are the minimum distance between two groups, then the classifier function f must satisfy $f(x_i) = \pm 1$ for the support vector data points. For other data points, $f(x) > 1$ if x is a value other than $(1: m, 1: n)$.For the linear support vector machine, if we define the classifier as $f(x) = w_i \cdot x + b$ and assume that two sets of data points with two different labels, $y_i = -1$ and $y_i = 1$, then the region between two groups is called the margin. The hyperplane is the area in the middle of the margin. We maximize the distance between these two sets. Therefore, $f(x_i) \geq 1$ for $y_i = \pm 1$ is a constraint to minimize $||w||$. Since $\frac{1}{||w||}$ is the distance between the classifier and the nearest data point, which is also called the support vector. In cases, where the data are not linearly separable, we want to minimize $\frac{1}{n} \cdot \sum max\big(0,1 - y_i(w \cdot x_i - b)\big) + \varepsilon ||w||^2$, with respect to ε in such a way that x_i lies on the same side of y_i. We collected two sets of instances through wavelet and short Fourier transform with two different labels $(awake = 1, drowsy = -1)$. Then we applied SVM with linear kernel to HRV. When the number of instances for binary classification is relatively small, generating the probability distribution function is impedingly difficult. Therefore, we aim to recognize and generate the pattern in minimum evaluation time in our study. In this case, the data can be classified by feature space similarities or differences.

By measuring the K-Nearest Neighbor (KNN), pattern recognition is achieved. The K neighbor label is applied to the new data. As K increases, the error rate decreases. Euclidean distance and Manhattan distance are two of many methods that can be used to measure the distance between the new data and its nearest neighbor. In this model, (x_i, y_i)denotes a training set of n labeled instances, where x_i belongs $R^d$and $y$belongs$(0, 1)$. We calculate

$$D(x_i, y_i) = ||L(x_i - y_i)||$$

Where $L = \sqrt{\sum (x - x_i)^2}$, and then we choose the K smallest D and according to these K numbers, we match the new instance to the majority label of these K inputs. K is an odd number. Therefore, there will be a majority for one label. In this case, we choose one nearest neighbor and Euclidian distance.

4.1Algorithm

Our research goal is to minimize the algorithm's training time period in order to maximize the system's accuracy. Since the amount of data collected is small, trying different machine learning techniques is necessary to achieve maximum accuracy. We select all features in both frequency and time domain to increase the system efficiency and optimize the algorithm. The ratio of low to high frequency decreases as the driver becomes drowsier. Therefore, the majority of the selected features are in the high and low frequency areas of the domain. In order to extract these features, we apply both wavelet transform (WT) and short Fourier transform (STFT). The result shows that using WT is better than STFT. This is due to the non-stationary nature of biomedical signals such as ECG. In wavelet transform, the algorithm applies varying size windows to the

HRV signal. The smaller window will generate higher resolution in high frequencies. Wider windows will generate higher accuracy in low frequency domains. Unlike WT, STFT has a less accurate frequency response due to the fact that the window size is constant.

According to our study, the accuracy improved by 5% when we replaced STFT with WT. We collected EKG samples from test subjects, filtered out the noise, and then the algorithm read the R-to-R peaks to form a new time series. The time series consists of the time difference between two consecutive R peaks and represents heart rate variation (HRV). Researchers confirmed that HRV in frequency domain contains valuable drowsiness related information. This information is correlated with the status of the autonomic nervous system (ANS). In the frequency domain, there are four major areas with the highest energy concentration namely as ultra-low frequency (UHF), very low frequency (VLF), low frequency (LF), and high frequency (HF). Low frequency range is between 0.02 and 0.06 Hz and high frequency range is between 0.12 and 0.4 Hz. In the past, the ratio of the maximum amplitude in the low frequency area to the maximum amplitude in the high frequency area was used as an indicator for drowsiness. This method usually required two minutes to collect the data, compute the ratio, and evaluate the drowsiness level. Moreover, the average accuracy level of this method was around 60% due to the noise and system errors. Our goal is to improve accuracy and lower the detection time by applying machine learning and selecting features from the low and high frequency regions. By applying different machine learning techniques, we improve the accuracy and reduce the detection time to less than thirty seconds. In this research, we extract 24 features in the frequency domain including peak, rms, area, and range. We apply the data to the support vector machine (SVM) and K-nearest neighbor (KNN) techniques. The result is above 85% accuracy.

4.2 Results

We ran multiple experiments on twenty-five subjects (11 females and 14males) ranging from ages 20 to 60 in both awake and asleep states. The HRV data was evaluated from a 3 leads ECG detector while the subjects were awake and drowsy for 8 hours straight. After applying an adaptive filter to remove noise from the signal, there was approximately 6000 HRV data for each experiment. In the next step, we extracted 32 features including 24features in frequency domains from both the LF and HF regions. To clarify, these two feature sets include maximum peak, range, standard deviation, peak to rms ratio, mean, skewness, kurtosis, energy, peak to amplitude of50% range, median inter-quartile range, median, edit distance on real signal, maximum to minimum distance, seqperiod, minimum, energy of right side of maximum, energy of left side of the maximum, the second largest element, the distance between the maximum and the second maximum and rssq in frequency domain. Furthermore, to improve the accuracy results, we added four arbitrary statistics in the time domainin cluding maximum, minimum, standard deviation, and energy. Also, we labeled the HRV data with “sleep” or “awake” with respect to the subject’s movements in order to preclude the drowsiness stage. Broadly, our results indicate that there is approximately ten-minute duration for the transition from awake to sleep state; in the subsequent step, epochs formed with a minimum range of 2 HRV samples to 128 HRV samples in each group. Moreover, to extract features in the frequency domain, two different wavelets and STFT methods were applied to the epoch groups in the LF and HF regions. We applied two different machine learning methods, SVN and KNN, for two different features extracted in wavelet and STFT. In short Fourier transform, we selected the fixed size windows and moved them along the HRV signal. Each window overlapped with the next window for half of the window size. The reason for the overlap is to reduce the effect of tapering that would result in the leakage at the boundaries. In addition, we chose the hemming window to reduce more of the leakage’s effect.

In the wavelet technique, the results were averaged in each time window over all of the epoch groups from 2 to 128 HRV bundles in order to increase the accuracy of computation. Afterward, we applied Morlet wavelet function with a continuous wavelet technique. Our study confirms the improvements in accuracy compared to those obtained from STFT. According to table 7**,** the accuracy of the algorithm improves over 2.5%after the features were extracted from wavelet analysis. However, we have added four features from the time domain to both of the analysis in addition to the extracted frequency features from wavelet and STFT. The results suggest that the KNN technique has a slightly higher accuracy in both wavelet and STFT analysis. This is due to the fact that KNN is known to classify binary outputs with a high accuracy even if the amount of collected data is low. There are comparisons between 80grouped epochs split in (80%,20%) for training and testing purposes. The results prove that the SVM and KNN techniques perform well over 80%accuracy despite the fact that the numbers of samples are low. We also increased K, the number of neighbors to increase the accuracy and noticed that the accuracy increase is much slower than the increase in the number of neighbors. As shown in table 8, there was a 1% increase to the system's accuracy when we increased the number of neighbors to 3. Also, changing the distance from Euclidian to other methods (e.g. the Minkowski distance technique) did not help to increase the accuracy as our study has shown. The Euclidian distance delivers the highest accuracy. In the SVM method, we tested 4 different kernels: cubic, Gaussian, linear, and quadratic. The results show that the highest accuracy is achieved via the quadratic kernel. However, it only improves by 2% over other kernel methods tested as highlighted in table 9 in this study.

Table 7. Average SVM- Wavelet- Confusion Matrix

N= 80	Predicted No	Predicted Yes	Total
Actual No	TP=26	FP=4	30
Actual Yes	FN=4	TN=46	50
Total	28	52	80

Table 8. Average SVM- STFT- Confusion Matrix

N= 80	Predicted No	Predicted Yes	Total
Actual No	TP=25	FP=5	30
Actual Yes	FN=4	TN=46	50
Total	29	51	80

Table 9. Average KNN-Wavelet- Confusion Matrix

N= 80	Predicted No	Predicted Yes	Total
Actual No	TP=25	FP=5	30
Actual Yes	FN=2	TN=48	50
Total	27	53	80

Table 10. Average KNN-STFT- Confusion Matrix

N= 80	Predicted No	Predicted Yes	Total
Actual No	TP=24	FP=6	30
Actual Yes	FN=4	TN=46	50
Total	28	52	80

Table 11. Accuracy table

Average Accuracy				
	KNN		**SVM**	
	STFT	WT	STFT	WT
Male	85.5%	88.3%	83.9%	87.6%
Female	81.4%	85.7%	81.1%	82.5%

Table 12. SVM comparison

Average Accuracy				
	KNN		**SVM**	
	STFT	WT	STFT	WT
Male	85.8%	89.3%	83.9%	87.6%
Female	82.1%	83.7%	81.1%	82.5%

Table 13. KNN comparison

Average Accuracy				
	KNN		**SVM**	
	STFT	WT	STFT	WT
Male	85.5%	88.3%	84.2%	87.8%
Female	81.4%	85.7%	81.3%	84.6%

4.3Conclusion

Our study proves that ECG signals are performing much better due to the non-stationary nature of ECG signals. Furthermore, the research proves that KNN generates higher accuracy. We suggest to applying unsupervised technique such as hidden Markov to HRV and classify drowsiness versus alertness. This is due to the fact that collecting ECG data for training purposes have unknown drowsy or awake label. Therefore, it is recommended to apply unsupervised technique such as Markov and clustering to generate a high accuracy algorithm with a low amount of data.

Algorithm for drowsiness detection

***Input**: The raw ECG Signal (ECGdata, Sampling_Frequency).*

---

1. *Normalized_ECG = ApplyNormalization (ECG_{data})* //Normalizing the raw ECG signal. //

2. *Filtered_ECG = AdaptiveFilter(Normalized_ECG)* //Filtering the ECG signal to remove noise using LMSAdaptive filter. //

3. *RtoR = R_PeakToPeak(Filtered_ECG)* //Computing the R peak-to-peak time series. //

4. *HRV = Resampling(RtoR, Sampling_Frequency)* //Resampling the R to R time series at a sampling rate of Sampling_Frequency. //

5. *Instances[m:int(n/m)] = Epoch(HRV[1:n])* //Grouping HRV data to apply wavelet analysis; m is number of HRV instances. //

6. $[F_{w1}, F_{w2}, F_{w3}, \ldots, F_{wn}]$ = *WaveletAnalysis(Instances)* //Extracting frequency based features from HRV usingcontinuous wavelet analysis; Fw_1 to Fw_n are frequency domain features. //

7.$[Fs_1, Fs_2, Fs_3, \ldots, Fs_n]$=*ShortFourierTransformAnalysis(Instances)* //Extracting frequency based features fromHRV using continuous short fourier transform analysis; $Fs_1$to$Fs_n$are frequency domain features. //

8. *Kernel*= 'Linear' //using linear classifier. //

9. *Kfold=0* //splitting training and test data set and folded into s different sets. //

10. *Distance='Euclidian' //*using linear classifier. //

11.*Num_Neighbors=1* //Choosing one neighbor distance. //

12.**do**

13.*Kfold=Kfold+1*//Increasing Counter loop. //

14. Training_data = t% of m instances//Splitting the original data set into t% Training and (100-t)% test set. //

15. Testing_data = (100-t)% of m instances //forming a Testing data set.//

16. [WSS0, WSS1, …, WSSm] = Support_vecto_machine ([Fs1, Fs2, Fs3,…, Fsm],training_data, Kernel) //Computing the coefficients of the support vector machine model using the training data set for short Fourier transform analysis. //

17. *[WWS0, WWS1, …, WWSm] = Support_vector_machine ([Fws1, Fw2, Fw3,…,Fwm],training_data, kernel)*//Computing the coefficients of the support vector machine model using the training data set for wavelet analysis. //

18. *[WSK0, WSK1, …, WSKm] = KNN ([Fs1, Fs2, Fs3,…, Fsm],training_data,Num_Neighbors, Distance)* //Computingthe coefficients of the support vector machine model using the training data set for short Fourier transform analysis. //

19. *[WWK0, WWK1, …, WWKm] = KNN([Fws1, Fw2, Fw3,…, Fwm],training_data, Num_Neighbors, Distance)*//Computing the coefficients of the support vector machine model using the training data set for wavelet analysis. //

20.*Accuracy_ShortFourierTransform(Kfold)=||Actual_output(Tesstindata)-Prediction(support_vevtor_machine(WSS0, WSS1, …, WSSm ,Testing_data))*//Evaluating accuracy of the linearsupport vector machine for extracted short Fourier transform features .//

21.*Accuracy_WaveletTransform(Kfold)=||Actual_output(Testing data)-Prediction(support_vevtor_machine(WWS0,WWS1, …, WWSm ,Testing_data)||* //Evaluating accuracy of the linear support vector machine for extracted waveletfeatures. //

22.*Accuracy_ShortFourierTransform(Kfold)=||Actual_output(Tessting data)-Prediction(KNN(WSK0, WSK1, …,WSKm ,Testing_data)||* //Evaluating accuracy of the linear support vector machine for extracted short Fouriertransform features. //

23.*Accuracy_WaveletTransform(Kfold)=||Actual_output(Tesstingdata)-Prediction (KNN(WWK0, WWK1, …,WWKm ,Testing_data))*//Evaluating accuracy of the linear support vector machine for extracted wavelet features. //

24. ***while*** *(Kfold!=5)* //Quitting the loop after 5 folds. //

25.*Final_Accuracy_ShortFourierTransform=average (Accuracy_ShortFourierTransform()) // calculating finalaccuracy for extracted short fourier transform features. //*

26.*Final_Accuracy_wavelet=average(Accuracy_ShortFourierTransform ()) //* calculating final accuracy forextracted wavelet transform features. //

5. Chapter 5. Unsupervised Machine Learning and Clustering Method

Drowsiness detection prevents human operation failures. Developing automatic systems with the least human interference such as drowsiness helps to increase system performance. Shin, Jung, Kim, and Chung (2010) stated that the accuracy and real-time process leads to lowering the number of car accidents and possible injuries and fatalities. As we mentioned in the introduction, several different algorithms have been developed and proposed to detect drowsiness. In particular, many algorithms study human electrical signals, such as EEG and ECG. In this chapter, we propose a novel method based on the ECG signal.

Applying a novel unsupervised machine learning clustering method to measure drowsiness is achieved without any preexisting ECG data. In supervised machine learning, the labeled pre-existing data established the algorithm. Therefore, for each object, the pre-recorded data must be collected before the algorithm is established. Kannan, Ramaswamy, Gujjar, and Bhaskar (2017) stated that the level of drowsiness in this algorithm is pre-set, but the level of drowsiness for each person changes due to factors such as age, environment and health. These factors vary over time. Therefore, a dynamic algorithm that actively measures the drowsiness with the different effects must be developed. Accurate drowsiness detection, followed by intervention, in drivers and heavy machine operators could someday help save lives and unnecessary expenses. The main issue in hardware design is the existence of noise. Sensors in a mobile environment produce data that is heavily polluted with environmental noise. The major goal in any hardware and software design is to remove noise and generate ECG free of noise for the drowsiness detection process, because noise is non-stationary. Some noise survives in hardware and software filtering, and the algorithm will detect and remove outlier data due to noise and signal distortion.

. If there is not any pre-existing ECG training set, then the algorithm would not know whether the person will be either asleep or awake at the time of testing. The raw ECG data does not have any information with which it can determine the level of drowsiness. Therefore, the data are unlabeled, and unsupervised machine learning is suggested at this point. Since the detection system consists of hardware and software, noise is the main challenge in detecting the correct level of drowsiness. Noise can negatively and severely impact the output of the reading if it is not eliminated during processing. Furthermore, collecting pre-recorded data for each subject is very difficult because the data has to be collected in two states: drowsy and alert, and it is very difficult to collect this data, especially for drowsiness. Moreover, the drowsiness level changes from person to person. Even in a single subject, the drowsiness level varies over time. For example, as a person gets older, his or her drowsiness level changes. The environment can also alter the level of drowsiness. To develop a system with minimized detection time and a maximized accuracy, it is imperative to remove all noise and anomalous data to minimize the processing time. Therefore, a dynamic algorithm constantly updates the level of drowsiness measurement for each person and does not require any pre-recorded data. At the same time, it will increase the processing time as it updates itself continuously.

Gomez (2014) formulated the n-dimensional space partition in d grids. Park et al. (2004) proposed a density clustering algorithm that monitors the data point in features dimensions space and inserts them into a pre-defined grid in the space. It will split the grid if each grid has high density. This iteration will continue as the smallest grid generated and it is called a unit cell. A cluster is a set of adjacent cells. Zhao et al.(2005) tested a method for feature extraction by using wavelet transform and applying SVM with a Gaussian kernel, but while the proposed approach attained an overall 98% accuracy, the method's complexity had a slow processing time. They proposed a new algorithm based on sliding and damping a window along a stream of temporal data and correlation between these windows statistics over time. However, they do not discuss the effect of noise and removing outliers on their algorithm. Xiao, Yan, Song, Yang, and Yang (2013) categorizes the sleep stages as rapid eye movement (REM) and non-rapid eye movement (NREM) and classifies the HRV data with random forest technique and achieved 88% accuracy. However, this method needs high number of features in frequency and time domain and it automatically will increase the chance of overfitting. Drowsiness is interpreted as the transition from alertness to REM and from NREM to alertness and vice versa. In Clustering time series using Unsupervised-Shapelets, Zakaria, Mueen, and Keogh (2012) introduced the new unsupervised method to cluster time series with distance calculated from the whole set using the Euclidian distance or Dynamic Time Wrapping distance. Silva, et al. (2013) proposed two major components of data stream clustering. In the online component, the data segment is generated with different methods such as windowing. Next, in the offline state, different clustering methods such as K-mean are applied to the data extracted from the window. They also discuss temporal aspects of data clustering since clusters get time-stamped, and he discusses the effect of detecting and removing outliers with clustering. Raudys et al. (1991) explained the effect of a small size training set for pattern recognition and how it affects the system accuracy. They discussed the effect of high dimensionality (number of features) and choosing the window size for density estimation. We used this method to evaluate the size of the window and the number of features. Chawla et al. (2014) studied the imbalanced data sets and they analyzed the tendency of classification to overfit. Jain (1997) measured the effect of extracting a high number of features from small size data and measured and compared the over fit effects for the small training sets.

In previous researches, the recorded data have been analyzed and algorithms were developed for existing data. However, all of these researches have been done off-line and does not analyze continuous data as it is received. In the real world, a driver's ECG output changes over time due to factors such as the length of driving and environmental interference. Moreover, noise has a dynamic destructive effect over ECG data reading and it has non-stationary nature over time. Therefore, a proper research to address these issues and to develop an effective algorithm is necessary. In this study, we suggest a robust method to analyze ECG data continuously over time, update the measurement of the level of drowsiness every few milliseconds, and remove the noise. This algorithm introduces a new online clustering and noise removal technique. Our proposed approach:

- Updates the algorithm continuously over time,
- Updates the level of drowsiness over time,
- Does not require any pre-recorded data, and
- Removes existing noise.

In this study, the algorithm continuously reads ECG data and updates the drowsiness evaluation accordingly. Also, the new data takes precedence over old data and eventually the older data fades away over time. Moreover, the algorithm detects the level of drowsiness without any pre-recorded data. Since the level of drowsiness is different from person to person and even time to time, the algorithm detects any changes in the level of alertness and does not need any pre-defined drowsiness level known to algorithm.

5.1 Methods

ECG output is a non-stationary data stream. Therefore, the data classification must be processed in real-time. Due to the increasing volume of data over time, the process can not include data from the beginning and there is not enough time for any offline process, because of both the limitation in memory space and the increasing workload for the processors. The stream of data is not labeled, and the detection system has no prior training set to establish the algorithm. Another major issue is that the drowsiness thresholds are different from person to person. Therefore, the new algorithm is designed to address these issues, by processing and classifying the collected ECG data in real-time. In this study, we form HRV data from ECG signals by measuring the time difference between two consecutive R peaks. This process forms a time series that represents the time difference from each R peak to the next R peak at the time the R peak is measured. The time series is unevenly measured in time since the R-peak intervals are not equal. This requires running an interpolation process to generate a time series in equidistant time intervals such that the Fourier transformation produces the HRV frequency response for further analysis. Since the stream of data is a one-time pass in temporal order, the algorithm divides the process into two parts. First, it applies a sliding weighted window to the current and past data and then emphasizes that the new data has a higher weight. Therefore, the importance of the older data fades out as the window captures the newer data. In the second stage, the offline process forms clusters, and the evolving of the cluster densities is measured, and these changes represent the alertness and drowsiness of the subject.

In this research, N features are extracted from the HRV signal in time and frequency domain (most features are extracted from LF and HF area). N-dimensional space is formed, and the total space can be represented as S_i. Therefore, the space is represented by

$$S = S_1 \times S_2 \times \ldots \times S_N$$

If each dimension is partitioned to m_i, where i=1, …, N. Therefore, the data space S is partitioned into V hypercubes, where

$$V = \prod_{i=1}^{N} m_i.$$

If we assign number to each hypercube and define hypercube index. Then, hypercube indices are calculated using

$$HC\_index\ (i_1, \ldots, i_n) = \ i_n \cdot \prod_{k=1}^{n-1} m_k + i_{n-1} \cdot \prod_{k=1}^{n-2} m_k + \cdots\ + i_2 \cdot m_1 + i_1$$

Where i_k's are sets of coordinates for hyper cubes and integer numbers from 1 to m_i.

If the stream of bundled ECG data arrives at time, t_1, then N features are extracted from the most recent bundle, and these N features placed as a data point in V space. As a result, we set the density index of the corresponding hypercube to 1. In the next step, a density index is assigned to each hypercube. This density index decreases as the data ages. For example, if the hypercube received a N feature data at time T_1 and the density index is measured in T_2, the index decreases by the value of $d(x)=\alpha^{T2-T1}$, where $0<\alpha<1$and x belong vspace. At time t, the density of each hypercube is calculated by

$$D(m_1, m_2, \ldots, m_N) = \sum_{All of x's} d(x)$$

Where $m_i < x_i < m_{i+1}$.

The density of each hypercube will be updated over time, if the newest data appears within the limits of that hypercube. To update all hypercubes at the same time, the $O(N^N)$ calculation is required. The density of each hypercube is limited by

$$Maximum of D = \frac{1}{1-\alpha}$$

This is proved by adding all D's up to the time T as follows.

$$\lim_{T\to\infty}\sum_{t=0}^{T} D = \frac{1-\alpha^{T+1}}{1-\alpha} = \frac{1}{1-\alpha}$$

Since there are V numbers of hypercubes, it is expected that the average density for the whole space is $\frac{1}{V(1-\alpha)}$. Also, D_n and D_m are two arbitrary numbers, a fraction of $\frac{1}{V(1-\alpha)}$.

$$D_n = \frac{\kappa_1}{V(1-\alpha)}$$

$$D_m = \frac{\kappa_2}{V(1-\alpha)}$$

κ_1 and κ_2 are two arbitrary numbers, such that $0 < \kappa_1 < \kappa_2 < 1$. We choose these two numbers to optimize the cluster definition in our algorithm.

A new concept is introduced as a floating hypercube. A floating-hypercube is a hypercube centered at each corner of the existing m_i's hypercubes. For the simplicity of the algorithm, we assume

$$m_i = M \qquad (i = 1, \ldots, N)$$

This floating-hypercube is placed at the corner of each hypercube and simply measures the density of the hypercubes with which it overlaps. The dimension of this hypercube is N and the size of each side is 1. The density of floating-hypercube is simply determined by adding the density of the portion of overlapped hypercubes. This floating-hypercube is sliding along all corners to scan the entire space and measure floating hyper-cube densities for each corner.

$D_f = \sum_i \frac{DHC_i}{2^m}$, i ∈ {All hypercubes overlap with the floating-hypercubes and DHC density represents overlapped hypercubes}

The clusters are formed as the floating-hypercube is scanning the entire space, the rules are:

1. If the floating-hypercube density is greater than D_n, then all overlapping hypercubes are assigned membership in the same cluster.
2. If the floating-hypercube is less than D_n, then the cluster wouldn't add any of the new overlapping hypercube.

The density of the clusters D_c, is equal to the average of density of all the hypercube members and are defined in table 14. Any hypercube can change density categories among all three types, high, medium, and low over time.

Table 14. Cluster threshold definition

Condition	**Density**
$D_c \geq D_m$	high-density cluster (HDH)
$D_n \leq D_c \leq D_m$	medium-density cluster (MDH)
$D_c \leq D_n$	low-density cluster (LDH)

By placing all neighboring high-density or medium-density hypercubes together, a cluster is formed. If the density of the floating hypercube exceedsD_n, then all overlapping hypercubes are placed in the same cluster. Since the density changes overtime, the membership of these hypercubes to each cluster can be removed or even added back later on. As a result, the shape of the cluster changes as a hypercube is demoted or added to a cluster. Specifically, if the density of floating hyper-cube is $D_f > D_n$. An MDH can evolve back into an LDH over time, and this causes the cluster shape as well. These changes happen continuously to the hypercubes at the edge of the cluster. Since the data stream is continuously updated, the size of the sliding window determines the time intervals in order to update the clusters. According to Babaeian et al. (2016), the 20 HRV sample minimum is the minimum ideal group for extracting to represent HRV in the frequency domain and representing the four major frequency areas: ULF, HF, LF, and VLF. Therefore, the window contains the 20 HRV sample minimum. If the window size is less than 20 HRV, the informational data will be lost, and if it's greater than 20 HRV, the evolution of the clusters may be lost because the time slot would be too large to detect the evolution. Moreover, the memory size and processor speed are two restricting factors that determine the time gap. At each step, the algorithm applies a novel method to detect and remove noise, outliers, and system errors. Therefore, a great advantage of this algorithm is removing noise, system errors, and anomalous data. A cluster with low density and a small number of hypercubes can represent the noise and outlier clusters. These clusters contain only LDHs and MDHs. They also receive few data samples over a short period of time and never get upgraded to HDHs. These clusters are different from the clusters that contain HDHs and MDHs that are downgraded to LDHs, and they might receive more data and be upgraded back to MDHs or HDHs. Therefore, the algorithm is detecting clusters that only receive a small amount of data in a short period of time and removes these noise outliers. The clusters that are downgraded from HDH to MDH may be upgraded back to HDH, and they cannot be considered noise. In the noise and outlier detection algorithm, small, low-density clusters are considered outliers and removals of these clusters helps speed up processing and minimize memory usage. If N features are divided by d hypercubes in each dimension, then there are N^d hypercubes formed. If the algorithm is to detect each hypercube in each process time, measure the density, and decide whether or not it's part of a cluster. This dramatically increases the number of operations and requires a lot of memory. Therefore, the curse of dimensionality will affect the performance. If the small and low-density clusters are removed, the algorithm saves a lot of time and memory space in each iteration. In this algorithm, the evolution of cluster densities represents the level of drowsiness. When a subject becomes drowsier over time, the density of these clusters changes and this algorithm applies the evolution of clusters overtime to measure the subject's drowsiness. In order to measure drowsiness, we define two new indices namely as ICD and SI as

$$\mathrm{ICD}(i, t) = \frac{\sum_1^M D(t)}{M(t)}$$

Where D represents the density of hypercubes in cluster i at time t and M (t) represents the number of hypercubes in cluster at time t, and

$$\mathrm{SI}(t) = \frac{M(t)}{N^d}$$

Where N is the number of dimensions, d represents the number of grids and N^d is the number of total hypercubes. Furthermore, SI (t) represents the shape of the clusters at time t. Although these indices are time-dependent, the time-evolving clusters will change these two indices over time.

Our research confirms that the changes in these two numbers are correlated with the level of drowsiness in the subject.

5.1.1 Cluster Selection

At any given time t, if there are c_i clusters formed, where i=1,…,L, and L is the number of clusters, then the algorithm searches for the two largest clusters. First, the algorithm sorts out the cluster with respect to $\mathrm{M}_{\mathrm{c_i}}(\mathrm{t})$. However, the two largest clusters must pass an additional test to make sure these clusters are among the clusters with the highest density. In this test, clusters are sorted out with respect to $\mathrm{ICD}_{\mathrm{c_i}}(\mathrm{t})$, and the two largest selected clusters must have ICD's higher than the highest interquartile range of the ICD's series. Interquartile range is a measure of statistical dispersion which is equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles. It is a measure of median of a series and also measuring the median the upper and lower half of the series and splitting the series into four different segments. If any of the two largest clusters fails the test, the algorithm selects the next largest cluster and repeats the test. It stops after the two clusters pass the test. These two clusters are named $\mathrm{C_h}(\mathrm{t})$ and $\mathrm{C_l}(\mathrm{t})$. In this study, we measure the evolution of the ratios of $\mathrm{ICD}_{\mathrm{c_h}}(\mathrm{t})and\mathrm{ICD}_{\mathrm{c_l}}(\mathrm{t})$ and also$\mathrm{SI}_{\mathrm{c_h}}(\mathrm{t})and\mathrm{SI}_{\mathrm{c_l}}(\mathrm{t})$.We prove that these ratios are consistent with the level of drowsiness and generate a unique range of numbers for each individual participant.

Algorithm 1: Search for the two largest clusters

$\boldsymbol{[ch, cl]} = \boldsymbol{findcluseter}(\boldsymbol{c})$

1.**for i=1: L**

2. Sort c_i respect to M_{c_i}

3.Calculate and Form ICD_{c_i}

4.**end for**

5.**for k=1: L**

6. Choose the two largest c_h and c_l

7.**If**ICD_{c_h} and ICD_{c_l} are not among the highest interquile range

ICD_{c_i} series

8. **then**

9.choose the next largest c_k and replace c_l or c_h with c_k

10.**endif**

11.**end for**

5.1.2 Noise Detection

The collected data is polluted with noise in driver drowsiness detection due to the mobility of the car, the possibility of long driving or hardware restrictions. There are few stages in the hardware and software design to remove noise due to the fact that the noise is non-stationary and changing over time. After filtering, some noise survives to pollute the system and contaminate the data. Therefore, the algorithm must include a stage in which noise will remove from the data. To clean the anomalous data, a new algorithm was developed. The noise is detected in the clusters that receive a small number of epochs. We expect few highly concentrated clusters that represent the true data, but the anomalous data forms a cluster with low density and outlier data. These clusters stay low density and never get upgraded to high-density clusters over the period of time. A cluster is called a suspicious of noise cluster if it satisfies following two conditions:

1. It stays at LDH for a long period of time and never gets upgraded to MDH and HDH.
2. The number of hypercubes in this cluster is much less than the number of hypercubes in high-density clusters.

In order to measure the system performance, we define a new index as

$$\mathrm{ER(t)} = \frac{\mathrm{NHL(t)}}{\mathrm{N(t)}}$$

Where ER represents algorithm error at time t, NHL represents the number of clusters maintained MDH, and HDH status in delta t where Δt>> 30s.In our research, we chose $\Delta t =$ 3 minutes. The set does not include any cluster that are promoted or demoted from MDH and HDH to LDH and vice versa. In this study, experimentally, we observe only two clusters to qualify for this set (C_h and C_l). It was explained how to extract these two clusters in the cluster selection method. N(t) represents the total number of clusters at the time t.

Algorithm: Noise detection and removal

Input: Clusterc_j, and number of clusters J, Time t, number of hypercubes in each cluster N_C

Output: Remove Clusters which represents noise, number of noisy clusters $\mathrm{N_O}$, Label formed clusters as “High Density”, “Medium Density” and “Low Density”.

[N_O, N_{HL} ,flag(c)]=Noise_Removal(c, J, t, N_C)

1.for k=1:J

2.$D_c(k)$ =$\sum_{n=1}^{N_C(k)} DHC(n)/N_C(k)$// Calculate the density of eah cluster.

3. **If**$D_c(k) \geq D_m$ then

4. set flag(cluster(k,t))=”HDH”

5.$N_{HL}(t) = N_{HL}(t) + 1$

6. **elseif**$D_n \leq D_c \leq D_m$ then

7. set flag(cluster(k,t))=”MDH”

8. $N_{HL}(t) = N_{HL}(t) + 1$

9.**else**

10. set flag(cluster(k,t))=”LDH”

11. **if** (flag(cluster(k,(t-6t_0):t)))=”LDH”) //flag

“suspicious” LDH clusters if they maintain this status for a

long period.

12. **if**($N_{c_j} < Q1 - 1.5IQ$) // Q’s are the interquartile of

number of hypercubes in clusters//.

13.$N_O(t) = N_O(t) + 1$

14: Delete c_j

15: **endif**

16: **endif**

17:**endif**

18: **end for**

For the first step, a binary flag is added to each cluster and if it is suspected of being noise, then it is checked periodically over two minutes. If it continues to be suspicious for a long period, then the cluster will remove. A new technique is used to evaluate if the number of hypercubes in a specific suspicious cluster is less than the number of hypercubes in the high-density cluster. This algorithm rules out that the number of hypercubes in the suspicious cluster is considered much less than the number of hypercubes in a high-density cluster at the time of t and L number of high-density clusters where each contains (N_1, …, N_L) number of hypercubes. In this algorithm, the rule of 1.5 interquartile ranges is applied as the number of hypercubes in a suspicious cluster is an outlier to the dataset of (N_1, …, N_L). We divide the (N_1, …, N_L) series into four groups called, from lowest to highest, Q1, Q2, Q3, and Q4. Each group represents 25% of the series. The range of the second and third group is called the interquartile (IQ) range. Any number below minimum (Q1)-1.5·IQ is called an outlier. Therefore, if the number of hypercubes in a suspicious cluster falls into an outlier group, the second condition is satisfied.

This will save a lot of processing time and memory in the processing of calculations. The time interval of each iteration is another critical factor. If this time period is too short, the accuracy of the calculation increases. However, the system will slow down due to the increased data volume. If the time interval is too long, a lot of valuable data will be lost. As a result, we will lose the accuracy. In the ECG process, Babaeian et al. (2016) found the minimum time in which drowsiness can be detected is25 seconds. If the detection time is less, a lot of valuable information will be missing due to the restriction in frequency response. Therefore, a time interval between 20 and 30 seconds will optimize the process.

Algorithm for drowsiness detection using ECG data

***Input**: The raw ECG Signal (ECGdata,*

Sampling_Frequency) and choose 20 s<t_0<30 s

***Output**: Level of drowsiness*

1.**while**(t=t+ t_0)

2.Normalized_ECG = *ApplyNormalization(*ECG_{data}*)*

//Normalizing the raw ECG signal.//

3.Filtered_ECG = *AdaptiveFilter(Normalized_ECG)*

//Filtering the ECG signal. //

4. RtoR = *R_PeakToPeak*(Filtered_ECG) //Computing the

R peak-to-peak time series.//

5. HRV = *Resampling*(RtoR, Sampling_Frequency)

//Resampling the R to R time series at a sampling rate of

Sampling_Frequency.//

6. $Instances_{[m:int(n/m)]}$ = *Epoch*($HRV_{[1:n]}$) //Grouping HRV

data to apply wavelet analysis; m is number of HRV instances.//

7. $[F_1, F_2, F_3, \ldots, F_n]$ = *WaveletAnalysis*(Instances)

//Extracting frequency-based features from HRV using

continuous wavelet analysis; F_1 to F_n are frequency domain features.//

8. $[F_1, F_2, F_3, \ldots, F_n]=[F_1, F_2, F_3, \ldots, F_n]/max[F_1, F_2, F_3, \ldots, F_n]$ // Normalize the extracted features. //

9.$S = S_1 \times S_2 \times \ldots \times S_N$//Form N dimension space.//

10.$V = \prod_{i=1}^{N} m_i$//Ndimensional space partitioned in d grids.//

11.$[F_1, F_2, F_3, \ldots, F_n] \rightarrow V$//Insert each feature data point in V space.//

12.$D(m_1, m_2, \ldots, m_N) = \sum_{T=t_0}^{t} \alpha^{T-t_0}$//Update the density of corresponding grid.//

13. $D_n = \frac{\kappa_1}{m^N(1-\alpha)}$ &$D_m = \frac{\kappa_2}{m^N(1-\alpha)}$ //where $0<\alpha<1$,κ_1, κ_2 Are arbitrary numbers and adjust with the hardware speed and memory size.//

14.$HC_{index}(i_1, \ldots, i_n) = i_n \cdot \prod_{k=1}^{n-1} m_k + i_{n-1} \cdot$

$\prod_{k=1}^{n-2} m_k + \cdots + i_2 \cdot m_1 + i_1$//Calculate the indices for each hypercube.//

15. **for**k=1:m^N//Set the floating-hypercube at the point (1, ,…, 1) in m^N dimension ,move and scan the entire space.//

16.**for**$i_k = 1{:}\, m_i$ //scan the rows.//

17.$D_f = \sum_{i_k} \frac{DHC_{i_k}}{2^N}$//Calculate the density of the floating-hypercube.//

18.**if**D_f > $D_L$then

Hypercube(HC_index) ∈ Cluster(J) //Add the membership of the overlappedhypercubes to cluster(J).//

19.**else**

Hypercube(HC_index) ∉ Cluster(J) //Revoke or ignore the membership of the overlapped hypercubes to cluster(J).//

20. J=J+1

21.**endif**

22. **endfor**

23. **endfor**

24.[N_O, N_{HL} ,flag(c))]=*Noise_Removal*(c, J, t, N_C)//Remove the noise clusters.//

25.$ER(t) = \frac{N_{HL}(t)}{N_O(t)+N_{HL}(t)}$

26. [ch, cl]=*findcluster*(c) //Find the two largest clusters(c_h and c_l).//

27. $ICD_{ch,cl}(t) = \frac{\sum_1^{M_{ch,cl}} D(t)}{M_{ch,cl}(t)}$ //Cluster Density Index.//

28. $SI_{ch,cl}(t) = \frac{M_{ch,cl}(t)}{N^d}$ //Cluster Shape Index.//

29. Density_Ratio(t)= $\frac{ICD_{ch,cl}(t)}{ICD_{ch,cl}(t-t_0)}$ //Compare these two indices with the previous window and evaluate and assess drowsiness if they are changing.//

30. Shape_ratio=$\frac{SI_{ch,cl}(t)}{SI_{ch,cl}(t-t_0)}$

31. **endwhile**

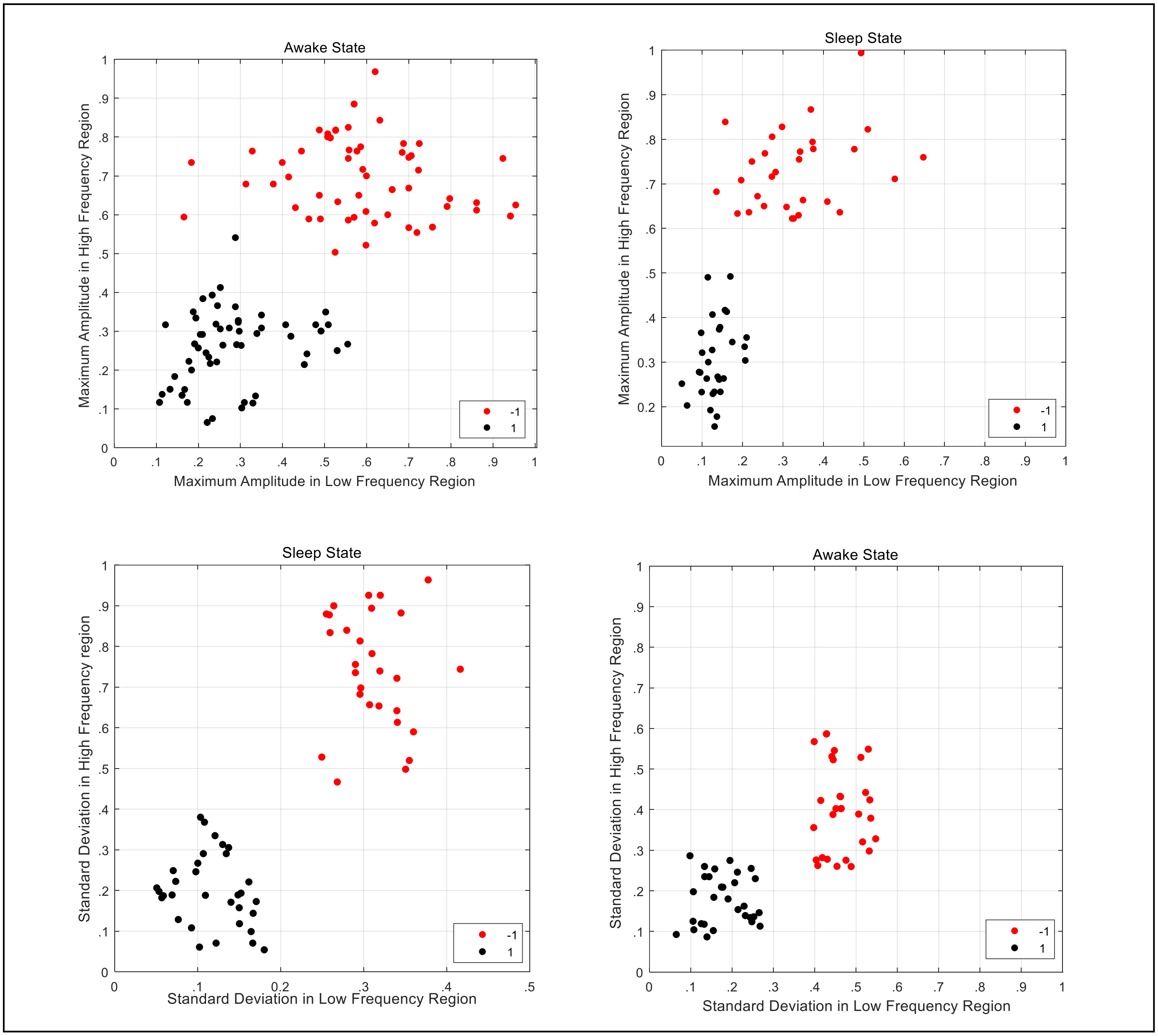

Figure 20. The clusters formed in two dimensions

Data was collected from 30 healthy subjects, aged 20 to 60, including 11 females. Subjects were recruited and taught to apply the three ECG leads. Subjects were instructed to connect the leads while awake and continue collecting data for four to eight hours as the subject fell asleep and until naturally awake up. The three leads were connected to a computer through the Arduino Uno board. Data was collected using Arduino software and stored on the computer hard drive. All subjects wear the Fitbit bracelet to verify that each subject enter each of the three sleep stages. The Fitbit bracelet application is able to monitor whether the subject is awake or asleep based on movement and heart rate. This verifies the integrity of the data. After collection, the noise is removed by adaptive filtering. Next, the HRV time-series signal is formed by measuring the distance between the consecutive R peaks in the ECG periodic signal. This time series represents the time difference between the peaks versus the time that they were measured.

In the HRV frequency domain, four features are extracted (maximum HF, maximum LF, standard deviation of LF, and standard deviation of HF) in each LF and HF area. These eight features in total form a multi-dimensional space. Each subject collected data during the complete sleep cycle including awake-sleep-awake cycles. Throughout these cycles, the participants were fully undisturbed and with minimal body movement to avoid additional noise to ECG data. We noted at that time that the subject is transitioning from awake to sleep and sleep to awake by smart device such as Fitbit and camera recording the eye blinking. In order to minimize the environment noise effects, adaptive filter was implemented and applied to ECG data. We were able to generate over 6000 HRV samples for each participant. In the next step, after resampling HRV data to generate the equidistance intervals and prepare the time series to apply frequency analysis, short Fourier transform (STFT) and Wavelet analysis (WT) were applied in 20 HRV sliding window to resemble the stream of data in the live mode. The sliding window includes 20

HRV data in each iteration, calculates the frequency response and at the last stage, extracts the eight features in the frequency domain for further analysis. These four features are collected in each region, HF and LF.

In the next step, eight and ten dimensions of space are formed. The features are maximum peak, range, peak to rms ratio, standard deviation, the second peak and energy. We studied the same cluster evolution and measured the index in eight and ten dimensions. The main goal in this research is to study the collected data for each subject, while we know the state of sleep, drowsiness, and awake at the time of the data collection. In this study, the cluster formation and evolution are processed through the proposed algorithm and confirms that it is consistent with the result of pre-measured label, sleep, drowsiness and awake. Therefore, two clusters are extracted from the algorithm process in different times of awake, drowsy and sleep. The ratio of number of $ICD_{c_h}(t)$and $ICD_{c_l}(t)$ and also $SI_{c_h}(t)$ and $SI_{c_l}(t)$ of these two clusters at the time of drowsiness, asleep, and awake are calculated. Asa result, these two ratios have been consistent over time with the changes of driver drowsiness. In the first run, we tested only maximum peak, and range for two regions, LF and HF. The algorithms immediately established clusters which are demonstrated in figure 20. Also, the density ratio of these two clusters consistently changes with the state of drowsiness over time; we represent these data changing in figure 21. We also measured the accuracy of the data by comparing with the actual data with the algorithm prediction output. The error at different times is less than 10% and the algorithm has the ability to measure the accuracy up to 90%. In figure 22, the density index is presented over then different time’s four features in two different states. The algorithm will detect the two highest density clusters and therefore, established the accurate estimate of drowsiness over time in 90%

of the time and proves how robust the noise detection technique in this algorithm filters out most of anomaly data and noise.

5.1.3 Results

The two drowsiness indicators ICD and SI confirm the consistency of cluster density and shape density changing with drowsiness as illustrated in figure 21. As the subjects become drowsier, the ratio of asleep ICD to awake ICD drops for all 30 subjects. However, the rate of change from awake to drowsiness is higher than the rate of change from drowsiness to sleep. Also, the rate of sleep SI over awake SI is changing consistently with the subject's drowsiness state. However, we noticed that this change is less noticeable than the density ratio indicator. We measured the error index which represents the ratio of noise interference. We also measured the error index for four- and eight-dimension space and noticed that the error index improves as more features are added to the dimensions. This result confirms that as the space are split to a finer grid, it gets easier to detect low density clusters and remove them. However, it adds more processing time and requires faster processor and higher capacity memories. In figure 23, we demonstrate the ER index for 30 subjects, and the graph shows the consistency of results.

5.2 Discussion

In this study, a novel method is presented to measure the drowsiness and applies an online clustering method. The advantage of this method is eliminating the need for any pre-existing measured ECG data and evaluating different levels of the drowsiness without any need for predefined drowsiness levels. Furthermore, the implemented noise removal technique detects noise online and removes it in the shortest time. As a result, the system accuracy is high and also reliable for further analysis such as other online health applications. Previous researchers have implemented many offline algorithms and they all require pre-recorded data to compare with the collected data. Therefore, their algorithms are unable to detect and remove noise if the noise is different from the noise detected at the time the pre-recorded data was collected.

The weakness of this algorithm is the speed of the system due to high dimension space (curse of dimensionality). The high-speed processor and higher capacity memories will increase the cost of the system. Therefore, this is a disadvantage of this system. Increasing the accuracy of the algorithm will have need of more features to build the higher dimension space and it compromises the cost of hardware. In short, the clustering technique will improve if the number of extracted features increases with respect to the available technology for a more powerful processor and memory with the fast and lease physical volume.

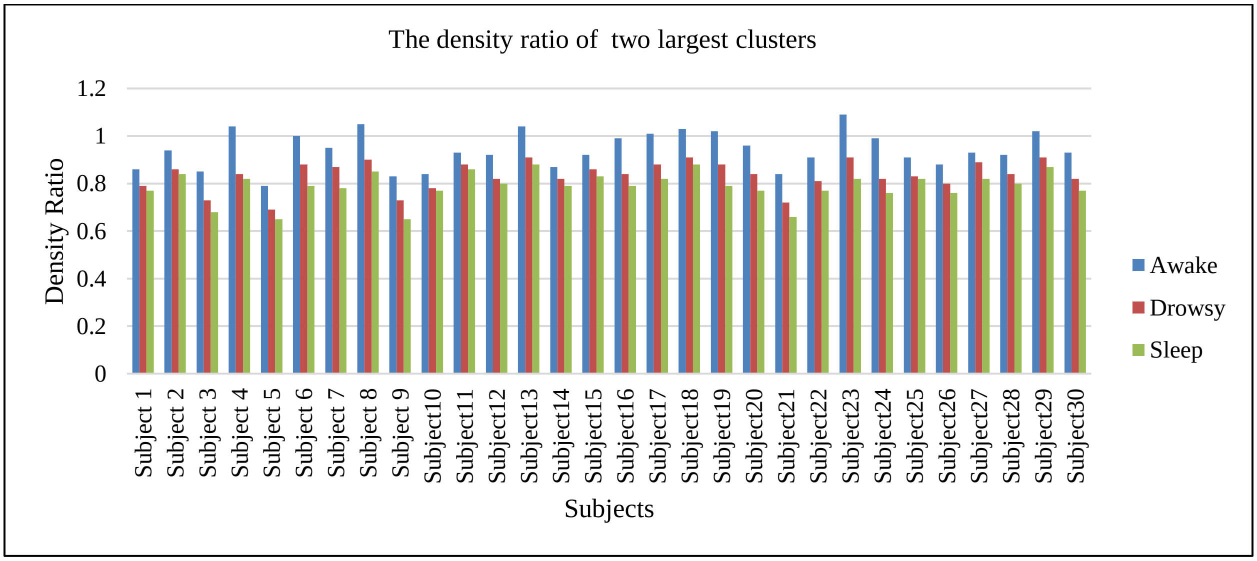

Figure 21. Density ratio of two largest clusters

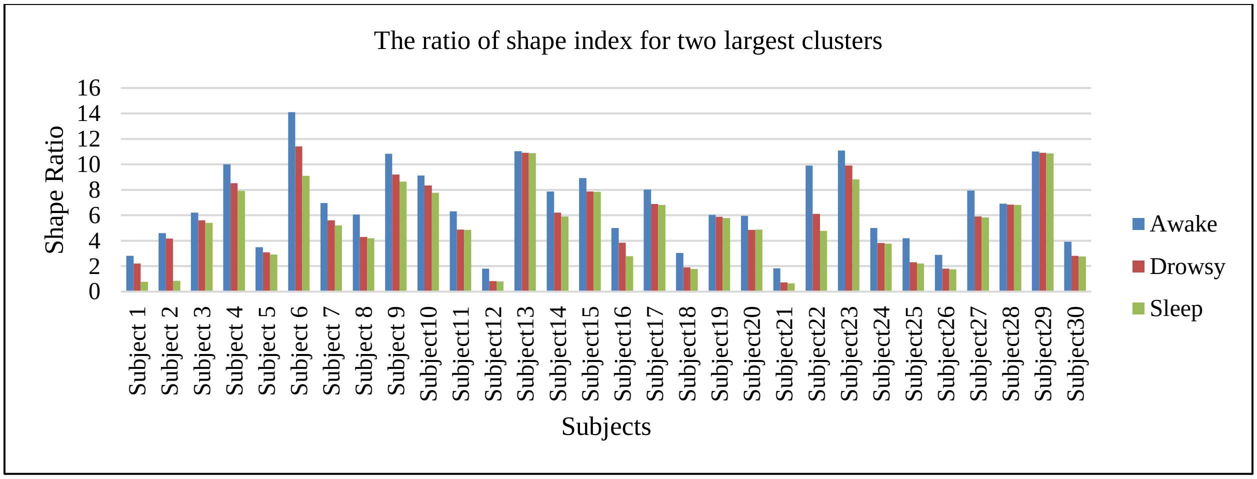

Figure 22. The ratio of sharp index for two largest clusters

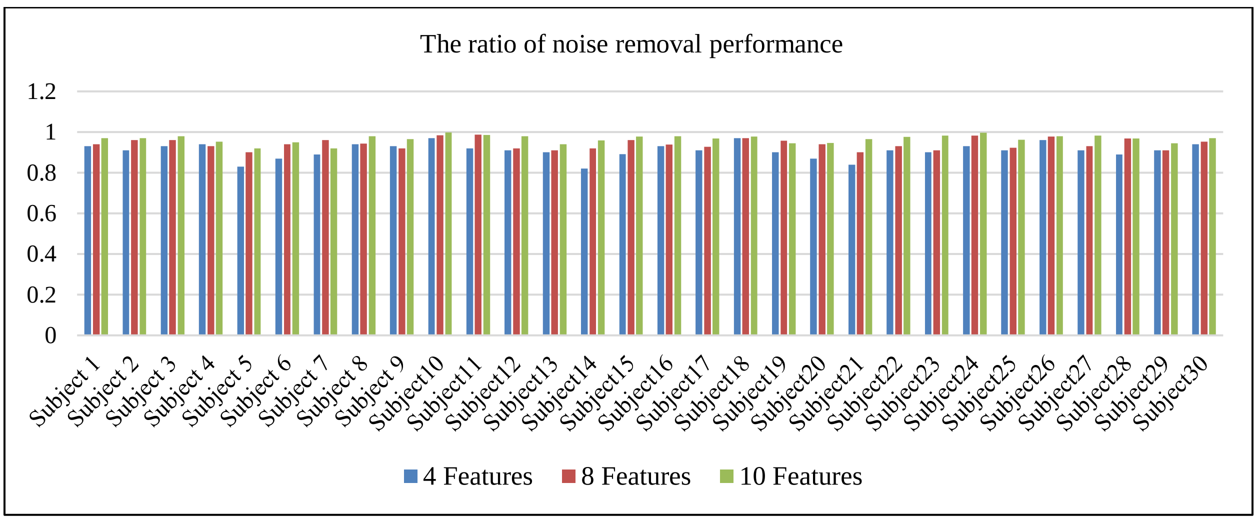

Figure 23. The ratio of noise removal performance

Bibliography

[1] Acharya, U.R., Joseph, K.P., Kannathal, N., Lim, C. M., & Suri, J. S.(2006). Heart rate variability: a review. *Med Bio EngComput,* 1031-1051.

[2] Assari A. M., & Rahmati, M.(2011). Driver drowsiness detection using face expression recognition.Paper presented at the IEEE International Conference on Signal and Image Processing Applications, Kuala Lumpur, Malaysia.

[3] Babaeian, M., Bhardwaj, N.,Esquivel, B. & Mozumdar. M. (2016). Real time driver drowsiness detection using a logistic-regression-based machine learning algorithm.Paper presented at theGreen Energy and Systems Conference (IGSEC), Long Beach, CA.

[4] Babaeian, M., Francis, K. A., Dajani, K., & Mozumdar, M. M. R.(2019). Real-time driver drowsiness detection using wavelet transform and ensemble logistic regression. *International Journal of Intelligent Transportation Systems Research,* 212-222.

[5] Baraniak, J., Hauer, J., Schuhmann, N., & Leugering, G. (2008). Implementation of Adaptive Filters for Biomedical Applications. Paper presented at the 8th International Conference on Computational Technologies in Electrical and Electronics Engineering, Novosibirsk, Russia.

[6] Baswade, A. M., &Nalwade, P. S.(2013). Selection of initial centroids for k-means algorithm.*International Journal of Computer Science and Mobile Computing*, 161-164.

[7] Borghini, G., Astolfi, L., Vecchiato, G., Mattia, D., & Babiloni, F. (2014). Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews*, 58–75.

[8] Bradley, P. S., Fayyad, U., & Reina, C.(1998). Scaling clustering algorithms to large databases. *KDD-98* Proceedings.

[9] Chawla, N. V., Japkowicz,N., & Kolcz, A.(2014). Editorial: special issue on learning from imbalanced data sets.*Sigkdd Explorations*, 1- 6.

[10] Chipman, M., & Jin, Y. L.(2009). Drowsy drivers: The effect of light and circadian rhythm on crash occurrence. *Safety Science*, 1364-1370.

[11] Deepa, T.P.& Vandana, R. (2016). EEG based drowsiness detection using mobile device for intelligent vehicular system.*International Journal of Engineering Trends and Technology*.

[12] Furman G. D., and Baharav. A. (2010). Investigation of Drowsiness while Driving: Utilizing Analysis of Heart Rate Fluctuations.*Computing in Cardiology*, 1091-1094.

[13] Ghosh, S., Nandy, T.,& Manna, N. (2015). Real Time Eye Detection and Tracking Method for Driver Assistance System.*Advancements of medical electronics*, 13-25.

Gomez, J. V. (2014). Generalyzing n-dimensional grid maps handling and neighbour cells extraction.*JV science and stuff*.http://jvgomez.github.io/fast_methods/ndgridmap.pdf

[14] Jain, A., & Zongker. D. (1997). Feature selection: evaluation, application, and small sample performance.*IEEE transactions on pattern analysis and machine intelligence*,153-158.

[15] Jen,K.-K.,& Hwang, Y.R.(2008). ECG Feature Extraction and Classification Using Cepstrum and Neural Networks.*Journal of Medical and Biological Engineering*, 31-37.

[16] Kannan, G. Ramaswamy, A. Gujjar, & Bhaskar, S. (2017). Drowsiness onset detection. ed: Google Patents.

[17] Karpagachelvi, S., Arthanari, M., & Sivakumar, M. (2010). ECG Feature Extraction Techniques - A Survey Approach. *International Journal of Computer Science and Information Security*, 76-80.

[18] Khazaee, A., & Ebrahimzadeh. A. (2010). Classification of electrocardiogram signals with support vector machines and genetic algorithms using power spectral features.*Biomedical Signal Processing and Control*, 252-263.

[19] Kumar, R.G., Samuel, V. P. R., & Santhosh, D.K.(2012).Classification of ECG signals for drowsiness detection in brain and computer interface.*GESJ:Computer Science and Telecommunications,* 1-12.

[20] Kuo, S. M., & Lee, B. H. (2001). *Real-Time Digital Signal Processing*. John Wiley and Sons Ltd.

[21] Kuronen, E.(2013). EPIC sensors in electrocardiogram measurement. (Unpublished master dissertation). Oulu University of Applied Sciences, Oulu, Finland.

[22] Laguna,P.,Moody,G.B., & Mark R.G. (1998). Power Spectral Density of Unevenly Sampled Data by Least-Square Analysis: Performance and Application to Heart Rate Signals,*IEEE Transactions on Biomedical Engineering*, 698-715.

[23] Langille, J. J. (2019). Human REM sleep delta waves and the blurring distinction between NREM and REM sleep.*The Journal of Neuroscience*, 5244-5246.

[24] Lee, K. M., Lee, S. M.,Sim, K. S., Kim, K. K.,&Park, K. S. (2009).Noise Reduction for Non-Contact Electrocardiogram Measurement in Daily Life.*Computers in Cardiology*, 493−496.

[25] Li, C., Zheng, C., & Tai, C.(1998). Detection of ECG Characteristic Points Using Wavelet Transforms. *IEEE Transactions on Biomedical Engineering,* 21-28.

[26] Li,G., & Chung, W.Y.(2013). Detection of driver drowsiness using wavelet analysis of heart rate variability and a support vector machine classifier.*Sensors,* 16494-16511.

[27] Lim, Y. G., Chung, G. S., & Park,K. S.(2010). Capacitive Driven-right-leg Grounding in Indirect-contact ECG Measurement.Paper presented at the32nd Annual International Conference of the IEEE EMBS, Buenos Aires, Argentina.

[28] Malik, S. (2004). Feature selection, L1 vs . L2 regularization. and rotational invariance. Paper presented at the Proceedings of the 41st annual Design Automation Conference, San Diego, CA, USA.

[29] Muhamad, A., Nasreen, B., Michael, D. (2017). A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability. *Sensors*, 1-16.

[30] National Highway Traffic Safety Administration's. (2017). *Traffic Safety facts*.https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812446

[31] National Highway Traffic Safety Administration's .(2018). *Research on Drowsy Driving*.https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812826

[32] Pagani, M., Lombardi, F., Guzzetti, S., Rimoldi, O., Furlan, R., Pizzinelli, P., Sandrone, G., Malfatto, G., Piccaluga, E. (1986). Power Spectral Analysis of Heart Rate and Arterial Pressure Variabilities as a Marker of Sympatho-Vagal Interaction in Man and Conscious Dog.https://doi.org/10.1161/01.RES.59.2.178

[33] Park, N.H., & Lee, W.S.(2004). Statistical grid-based clustering over data streams.*Sigmod Record*, 32-37.

[34] Patel, S. K. L. L. M., Cavanagh, D., Rossiter, P. (2011). Applying neural network analysis on heart rate variability data to assess driver fatigue. *Expert Systems with Applications*, 7235-7242.

[35] Patela, M., Lala, S. K. L., Kavanagha, D., and Rossiterb, P.(2010). Applying neural network analysis on heart rate variability data to assess driver fatigue.*Expert Systems with Applications*, 7235–7242.

[36] Raudys, S.J., & Jain, A.K. (1991). Small Sample Size Effects in Statistical Pattern Rwcognition: Recommendations for practitioners.*IEEE Transactions on pattern analysis and Machine Intelligence*, vol.13.

[37] Sahayadhas. A., Sundaraj. K, & Murugappan. M. (2012).Detecting driver drowsiness based on sensors: a review. *Sensors*, 16937-16953.

[38] Sang-Joong, Heung-Sub Shin & Wan-Young Chung.(2014). Driver fatigue and drowsiness monitoring system with embedded electrocardiogram sensor on steering wheel.*IET intell. Transp. Syst*, Vol.8.

[39] Sangeetha, M.S, Rajendiran, M.,&Malathi,G.(2015). Embedded ECG based real time monitoring and control of driver drowsiness condition. *International Journal of Science, Technology and Society*, 146-150.

[40] Seyd. P. T. A., Ahamed. V. I.T., Jacob, J., & Joseph, P. K. (2008). Time and frequency domain analysis of heart rate variability and their correlations in diabetes mellitus.*International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, 85-88.

[41] Shin, H.-S., Jung, S.-J., Kim, J.-J., & Chung, W.-Y.(2010). Real time car driver's condition monitoring system. Paper presented at the IEEE Sensors Conference, Busan, South Korea.

[42] Silva, J. A., Faria, E. R., Barros, R. C.,Hruschka, E. R., Carvalho, A. C.,& Gama, J. (2013).Data stream clustering, *ACM Computing Surveys*, 1-31.

[43] Swapnil, V. D., Bharti, T.A., Ansari, A.,& Khatib, M.S.(2014). Hardware implementation of driver safety system. *IOSR Journal of Computer Science*, 79-82.

[44] Sztajzel, J. (2004). Heart rate variability: a noninvaisive electrocardiographic method to measure the autonomic nervous system, *Swiss Medical Weekly*, 514-22.

[45] Tekade, P. M., & Gawali, S.(2012).Investigation and New Method of No intrusive Detection of Driver Drowsiness, *International Journal of Engineering and Innovative Technology*, 210-216.

[46] Thomas, K., Andreas, R., & Nikoletta, S. (2017). A robust drowsiness detection method based on vehicle and driver vital data. Paper presented at the Mensch und computer Conference, Regensburg.

[47] Tijerina, L., Gleckler, M., Stoltzfus, D., Johnston,S., Goodman,M.J., & Wierwille,W.W.(1999). A preliminary assessment of Algorithms for drowsy and Inattentive driver detection ontheroad. https://ntlrepository.blob.core.windows.net/lib/17000/17900/17991/PB2001105783.pdf

[48] Wierwille, W. W., Lewin, M. G., & Fairbanks. R. J. (1996). Research on vehicle-based driver status/performance monitoring. https://doi.org/10.1037/e460512008-001

[49] World Health Organization.(2015). *Global Status Research on Road Safety*. http://www.who.int/violence_injury_prevention/road safety_status/2015/en/

[51] Wu, X., Kumar, V., Ross Q., Ghosh, J., Yang, Q.,& H. Motoda.(2007).Top 10 algorithms in data mining,*Knowledge and Information Systems*, 1-37.

[52] Xiao, M., Yan, H., Song,J., Yang,Y., & Yang, X. (2013). Sleep stages classification based on heart rate variability and random forest, *Biomedical Signal Processing and Control*, 624-633.

[53]Xu, X., & Frank, E. (2004). Logistic Regression and Boosting for Labeled Bags of Instances. Paper presented at the 8th Pacific Asia Conference, Sydney, Australia.

[54] Xun, Yu. (2009). Real-time nonintrustive detection of driver drowsiness. (Unpublished master dissertation). University of Minnesota, Duluth, United States.

[55] Xun,Yu. (2012). Integrated Approach for Nonintrusive Detection of Driver Drowsiness. (Unpublished master dissertation). University of Minnesota, Duluth, United States.

[56] Yilmaz,B., Asyali, M.H., Arikan, E., Yetkin, S., & Ozgen, F. (2010). Sleep stage and obstructive apneaic epoch classification using single- lead ECG. *Bio Medical engineering*,1-14.

[57] Zakaria, J., Mueen, A.& Keogh, E.(2012). Clustering time series using unsupervised shapelets.Paper presented at the IEEE 12th International Conference on data mining.

[58] Zakharov, R., & Dupont, P. (2011). Ensemble logistic regression for feature selection. Paper presented at the 16th IAPR International Conference on Pattern recognition in Bioinformatics, Delft, the Netherlands.

[59] Zhao,Q., & Zhan. L. (2005). ECG feature extraction and classification using wavelet transform and support vector machines.Paper presented at the International Conference on Neural network and brain.