Real-time Stress Monitoring System

Akshay Bhardwaj, Kriti Singh, Purushothaman Saravanan, Ashwini Muralidharan, and Nalina Kizhakkekkara Vadakkoot

***Abstract*—Stress is a prevalent condition among drivers and is frequently associated with cardiovascular issues. The significance of this study lies in its potential to mitigate the adverse effects of stress on drivers - a factor closely associated with aggressive driving behaviors that often lead to accidents. An automated and streamlined system for detecting stress in drivers is essential and this research centers on the development of a real-time stress monitoring system that classifies ECG signals collected from a sensor, using an effective Deep Learning model based on the VGGNet architecture. The proposed classifier was trained on raw ECG data collected from drivers while they were driving along a predetermined route encompassing both city streets and highways. The VGGNet architecture achieved a total accuracy of . The comprehensive implementation of this system involved the development and integration of a real-time ECG sensor, carefully encased in a protective housing designed for secure attachment to the seatbelt. The captured ECG signal was subsequently transmitted to a Jetson Nano, where the inference process took place through the trained Deep Learning model. The results of the inference were then conveyed through a user-friendly frontend hosted on a Flask application.**

***Index Terms*—signal-processing, real-time, deep learning, stress monitoring, feature extraction**

I. Introduction

Stress is a prevalent phenomenon affecting individuals across different age groups and backgrounds. It represents a natural response to challenging circumstances. However, when it persists over time, it can contribute to various health issues such as depression, anxiety, and cardiovascular problems. Therefore, it is imperative to engage in stress monitoring as a preventive measure against these health concerns. Monitoring stress levels enables individuals to pinpoint stress triggers and develop effective strategies for their management [1].

Numerous factors can impact a driver's capacity to operate a vehicle safely. However, one aspect that often remains underemphasized is the individual's stress level. It is widely acknowledged that stress exerts detrimental effects on our overall well-being, and this extends to its impact on our performance while driving [1].

Against this background, an automated system for stress monitoring using electrocardiogram (ECG) signals is proposed. Recent advancements in wearable technology have made it practical to continuously measure various physiological signals with minimal disturbances. This project intends to use deep learning technology to automate ECG monitoring for stress detection. Deep learning offers several advantages over traditional methods for stress monitoring. They can capture subtle and intricate markers of stress that might be overlooked by conventional approaches. Moreover, deep learning can adapt and refine its understanding through continuous exposure to new data, enhancing its accuracy and adaptability over time. Deep learning enables the creation of highly personalized stress monitoring architectures and pipelines, offering insights that were previously inaccessible through conventional means.We train a deep learning model on an open source dataset so that it can learn the underlying pattern in ECG signal corresponding to different levels of stress and can classify the stress levels in new unseen data.We rely on our model to classify stress in real time by validating its performance on a subset of training data that was kept for testing.

Classifying stress levels from ECG signals using peak detection is a common approach. ECG signals exhibit distinctive peaks, such as the R-peaks (associated with ventricular depolarization) and other characteristic waves and intervals. Analyzing these peaks can provide valuable information for stress classification [1].

The rest of the paper is organized as follows. In Section II, we provide a comprehensive literature review encompassing various investigations concerning accidents associated with driver stress, as well as an overview of deep learning approaches for stress detection. Section III and Section IV describe the details of the dataset and data preprocessing that we propose to use. A brief overview of machine learning (ML) algorithms for stress classification is discussed in Section V. The proposed method is described in Section VI.

II. Related Literature

In the existing literature, a multitude of studies have explored the assessment of driving performance alongside the evaluation of drivers' physiological and cognitive states [2], [3]. For instance, in [2], researchers introduced a methodology to gauge a driver's relative stress level by employing physiological data analysis and artificial intelligence techniques. This study involved the participation of twenty-four drivers who were continuously monitored during real driving sessions lasting at least 50 minutes. The findings underscored a significant correlation between stress levels and skin conductivity and pulse metrics.

In another study, [3], changes in shoulder and neck muscle activity were analyzed while participants engaged in driving simulations, encompassing both professional and non-professional drivers. Notably, fatigue was observed in both groups after just 15 minutes of driving. This growing body of research has consistently established a robust connection between emotions and traffic accidents. For instance, researchers in [4] conducted a cluster analysis on survey responses from over 1500 college students, focusing on situations that could potentially lead to driving-related irritations. Their investigation revealed gender-specific anger triggers, with men expressing more frustration towards police presence and slow driving, while women were more inclined to be angered by illegal behavior and obstructions. Consequently, these studies highlight the potential utility of understanding driver emotions as a means to mitigate traffic accidents, as negative emotions such as fear, anger, disgust, or sadness have been shown to increase the likelihood of engaging in hazardous driving behaviors.

Several studies have explored the use of deep learning architectures for stress detection based on ECG signals. [5] propose a simple algorithm for the classification of ECG signals as stress or normal by the automatic detection of heart rate variability from R peaks through the discrete wavelet transform (DWT) method. Machine learning algorithms use various parameters obtained from classification to find the accuracy of the results. Deep neural networks were trained on the two publicly available stress classification datasets in [6]. The classification was done after applying pre-processing techniques, such as data pruning, down-sampling, and data augmentation, using a sliding window approach. [7] proposes multimodal machine learning techniques for stress detection on individuals using multimodal datasets recorded from wearable physiological and motion sensors. Data of sensor modalities like three-axis acceleration (ACC), ECG, blood volume pulse (BVP), body temperature (TEMP), respiration (RESP), electromyogram (EMG), and electrodermal activity (EDA) are for three physiological conditions - amusement, neutral and stress states, were taken from WESAD dataset [8].

III. Dataset

A dataset composed of a series of multiparameter recordings acquired from healthy volunteers while they were driving along a predetermined route encompassing both city streets and highways in the vicinity of Boston, Massachusetts was used in this study. This dataset is dubbed as the *DriveDB* dataset and is thoroughly described in [9]. The primary objective behind collecting this dataset was to explore the potential for automated stress recognition based on the recorded signals. These signals encompass ECG, QRS (right trapezius), and GSR (galvanic skin resistance) measurements obtained from the hand and foot, in addition to respiration data. This study uses only the ECG signal for stress prediction, a segment of which is shown in Figure 1. The dataset follows the lead II standard configuration to capture the ECG signal, which results in minimizing the motion artifacts and producing a rhythm trace with sharp R-waves. In order to convert the dataset into label-sample form, the marker signals that were embedded in the ECG signals by the authors were leveraged. The marker signal has distinguishable peaks that separate the ECG signal segments, which are annotated with different stress states. The annotations first had to be upsampled to match the frequency of the ECG signal (496 Hz). Due to the difficulty of those driving experiments, there were several errors and problems with the ECG and marker signals of some subjects, hence, they were omitted from this study. The resulting dataset included data only from Driver 6, 7, 8, 10, 11, 13, 14, and 15. The final annotated dataset had three stress levels being quantified as *Level 1* for low stress, *Level 2* for moderate stress, and *Level 3* for high stress level.

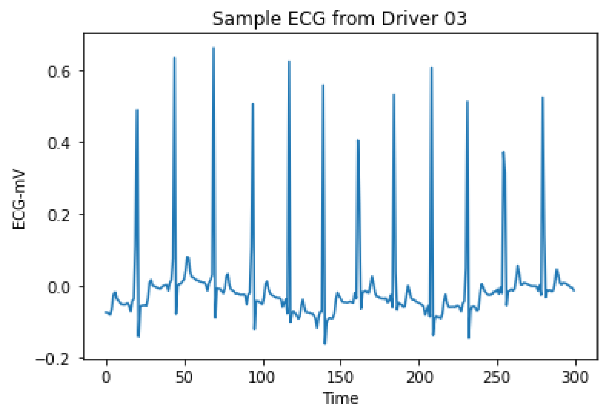


Fig. 1: A sample segment of ECG data

IV. Data Processing pipeline

The DriveDB dataset mentioned in Section III was used for training to models for predictive analysis. Raw ECG data is almost never used directly for predictive analysis due to noise, artifacts and other complexities, requiring preprocessing and feature extraction to extract relevant information for effective and accurate predictive modeling. Often, ECG signals are noisy, which makes it difficult to identify abnormalities and diagnose heart conditions. The most common factors that cause noise in ECG data are:

- Baseline wander: Slow, drifting movement of the baseline of the ECG signal, caused by small fluctuations in the electrical potential of the skin.
- Artifacts: Unwanted signals are not caused by the heart's electrical activity, and are caused by movement of the electrodes, loose connections, etc.

Preprocessing generally begins with applying filters. Filters can be used to remove these unwanted frequencies from the ECG signal while preserving the frequencies that are important for diagnosis. Raw ECG data was first flattened by removing baseline wander using a median filter. A median filter is a non-linear filter that replaces each element in an array with the median of its neighboring elements. This was used to smooth out the signal and remove baseline wander in an array, as illustrated in Figure 2.

The ECG signal contains frequencies from 0.05 Hz to 150 Hz, with the most important information contained in the 0.05 Hz to 40 Hz frequency band. Therefore, a bandpass filter with cutoff frequencies of 0.05 Hz and 40 Hz [10] was used to filter ECG signals. The filter removed noise below 0.5 Hz and above 40 Hz while preserving the important information in the ECG signal. The upper bound was chosen as 40 Hz for an ECG bandpass filter because there is no clinically significant

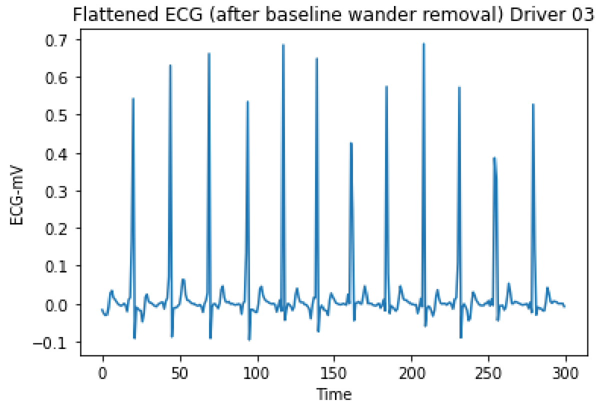


Fig. 2: Baseline Wander Removal

information in the ECG signal above 40 Hz. Frequencies above 40 Hz are typically associated with noise and artifacts and, hence, discarded. This is observed in Figure 3.

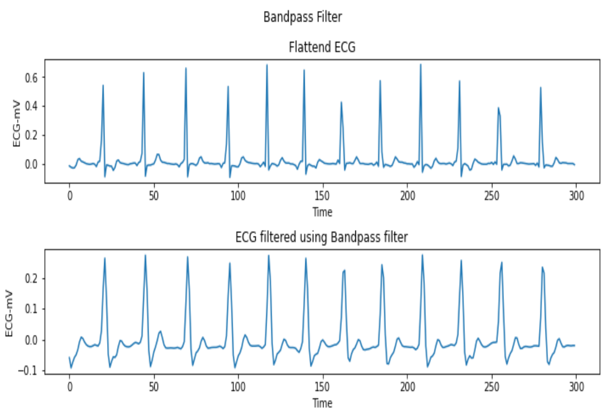
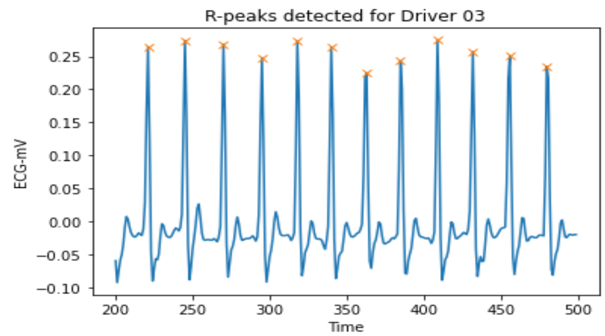


Fig. 3: A segment of flattened ECG vs Flattened ECG after Bandpass filtering

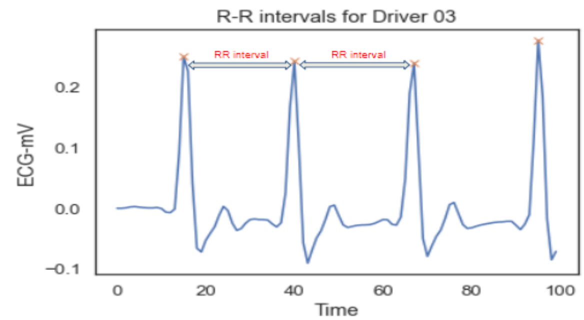
R peaks on an ECG represent the highest point of the QRS complex, indicating ventricular depolarization, and their accurate detection is crucial for assessing heart rate variability and diagnosing various cardiac conditions. Using a simple thresholding algorithm, keeping height and distance to other peaks as determining metrics, R-peaks of ECG were identified. The RR interval on an ECG represents the time duration between successive R peaks, reflecting the time between consecutive ventricular depolarizations. With the estimated R peaks, the RR interval between successive R-peaks was calculated by taking the difference between the time of successive R-peaks. This is illustrated in both Figure 4a and Figure 4b respectively. This was used to determine the Heart Rate (HR) using the formula

$$HR = \frac{60}{RR\ intervals\ in\ seconds}$$

The heart rate values were associated with corresponding ECG values through the identification of R peaks, and the



(a) R-peaks of ECG



(b) RR interval between successive R-peaks

Fig. 4: R-peak and RR Interval detection

remaining values were filled in using polynomial interpolation. This dataset with ECG and corresponding Heart Rate values served as input to the Deep Learning algorithm, discussed in Section V-B.

Machine learning models require additional processing to effectively capture temporal dependencies, necessitating further preprocessing steps such as segmentation and feature extraction. Hence, the data was segmented into windows of length 2 seconds. For each window length, it was stored as a 1-dimensional window of $w \times fs$ time-steps, where w is the length of the window in seconds and fs is the sampling frequency in Hz. Powerful feature extraction system is necessary it transforms raw data into a meaningful representation that can be effectively utilized by machine learning algorithms in stress detection and classification. Each segment of ECG and HR window was converted into 5 human-engineered features, as listed and defined in Table I. This concluded the data preprocessing and feature extraction steps, and the data was now prepared for input to the ML classifiers.

V. Stress Classification

A. *Machine Learning Algorithms*

The feature extractor interfaces with the classification module, responsible for conducting predictive analyses. Within the realm of ECG data analysis, conventional machine learning algorithms, such as K-nearest neighbors (K-NN), and Naive Bayes, have been widely utilized and documented in the literature. Additionally, tree-based algorithms like Decision Trees

Feature	Description
Feature Mean	The feature mean of a segment is the average value of the data points within that specific segment.
Feature Standard Deviation	The standard deviation of a segment is a measure of the amount of dispersion or spread of data points within that specific segment.
Skewness	Skewness is the asymmetry or lack of symmetry in the distribution of amplitude values across the signal in a segment.
Kurtosis	Kurtosis the peakedness or flatness of the distribution of amplitude values within the signal, indicating the degree to which the waveform deviates from a normal distribution.
Spectral Entropy	Spectral entropy is a measure of the randomness or disorder in the frequency distribution of a signal, providing insights into the diversity and distribution of frequencies within a given spectral range.
SDNN	SDNN measures the overall variability of the heart rate.It is calculated by taking the standard deviation of all the normal-to-normal (NN) intervals between consecutive heartbeats.
RMSSD	RMSSD is a time-domain measure that reflects short-term variability in heart rate.It is calculated by taking the square root of the mean of the squared differences between successive NN intervals.

TABLE I: Features Extracted

and Random Forest have demonstrated superior performance in certain cases. This study trained and deployed traditional ML Algorithms like K-NN, Decision Trees, Random Forest and XGBoost, the results of which are presented in Table II. While machine learning classifiers serve as a primary choice due to their lightweight nature and ease of training, they may fall short in adequately modeling temporal patterns [1].

B. Deep Learning Algorithms

Deep learning techniques are renowned for their ability to capture intricate non-linear relationships among various features within datasets. Consequently, the exploration of deep learning algorithms becomes imperative, particularly for handling time-series data and acquiring proficiency in learning temporal dependencies [1].

For this study, a VGGNet architecture was employed to effectively capture hierarchical features in the input data. The model structure was organized into four blocks, each incorporating a 1D convolutional layer, batch normalization, rectified linear unit (ReLU) activation, and max-pooling for downsampling. The number of filters progressively increased through the blocks $[64, 128, 256, 512]$, enabling the network to learn and extract intricate patterns. An adaptive average pooling layer ensured a consistent feature size, facilitating subsequent processing. The classifier, composed of two fully connected layers with ReLU activation and dropout for regularization, produced the final predictions for the three classes. Following the principles of VGG-like architectures, the model prioritized the use of small 3×3 convolutional kernels and max-pooling to enhance feature extraction capabilities [1]. Overall, the VGGNet architecture served as a robust framework for extracting and classifying complex patterns in the input ECG and HR data. The results of this architecture are presented in Section VII-B.

VI. Proposed Method

The proposed method involves the acquisition of real-time ECG data from specialized hardware, which is then transmitted to an NVIDIA Jetson central server. This will serve as the core processing unit, where the received ECG data undergoes preprocessing to enhance data quality. Subsequently, machine learning algorithms will be employed to classify the ECG signals into relevant categories, providing users with timely and accurate feedback on their stress levels.

A. Hardware Design

There are several types of sensors used for acquiring ECG data. Electrodes are attached to specific locations on the body and the electric potential differences between different electrode placements create the signal. Lead systems are used to define specific electrode placements to capture different perspectives of the heart's electrical activity. Cables and connectors are used to ensure accurate transmission of small electrical signals from the body to the recording device. Amplifiers are used to increase the signal strength and filters are implemented to remove the noise from the signal.

This Project's Hardware is divided into three modules: Power, Analog Front End, and WiFi. Figure 5 shows the power module. The board is powered by a 3.7V LiPo Battery. A step-down LDO is used to provide a constant 3.3V to the WiFi module and other op-amps. It also has a battery charging circuit.

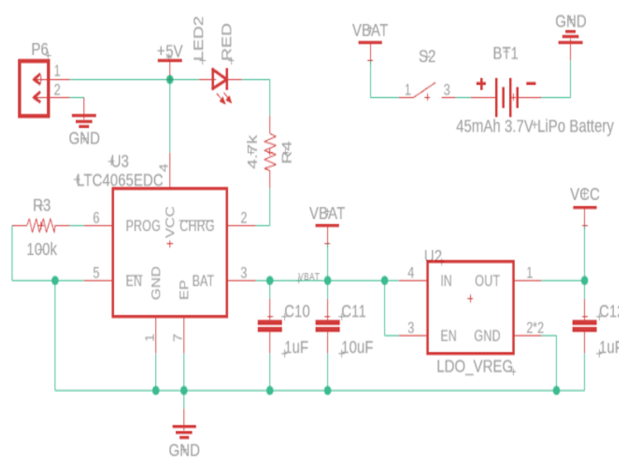
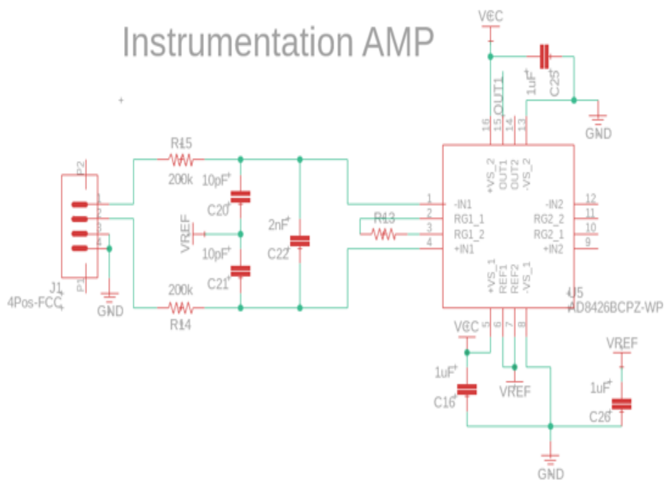
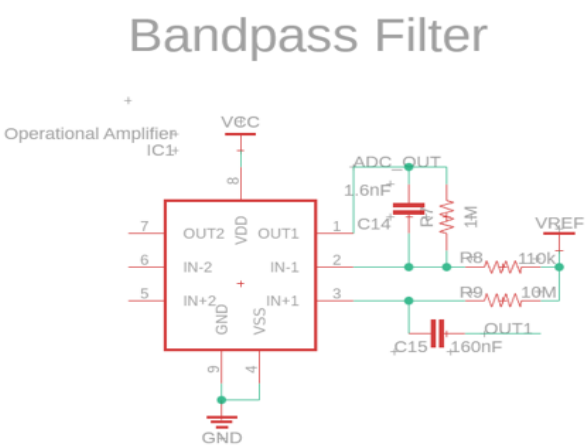


Fig. 5: Schematic of power module

The Analog Front end of the board is responsible for capturing and sending the ECG signal to the microcontroller. The instrumentation amplifier as shown in Figure 6a, amplifies the difference between two input signal voltages while rejecting any signals that are common to both inputs. The bandpass filter in Figure 6b, has a gain of 10 and a frequency range from 0.1Hz to 100Hz.



(a) Instrumentation amplifier



(b) Bandpass filter

Fig. 6: Schematic of Analog front end

ESP32 Wi-Fi module collects the ECG signals in real-time and is sent over to a server where the post-processing is done. Figure 7shows the schematic of the microcontroller. Figure 8.a shows the PCB layout of the device which was designed using Fusion 360 software. The following layout was exported and a 3D model was generated as shown in Figure 8.b. The PCB was then fabricated and the components were soldered and assembled as seen in Figure 8.c

The frequency response was taken using Digital oscilloscope. The following can be interpreted from the graph in Figure 9. The gain is 59.272dB and the high and low cutoff frequencies were 60Hz and 0.1Hz respectively.

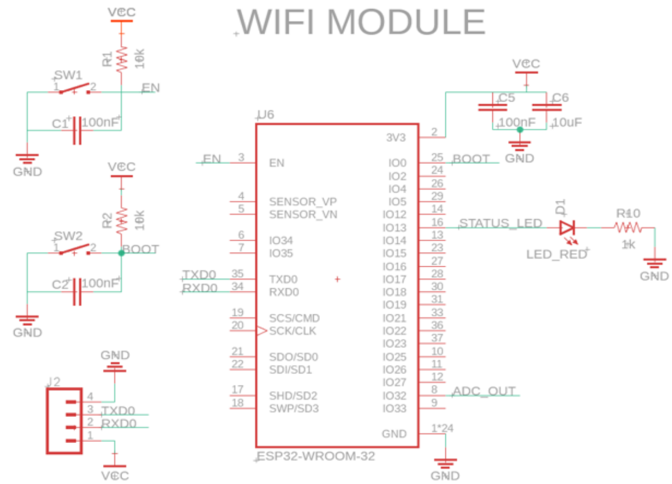


Fig. 7: Schematic of WiFi module

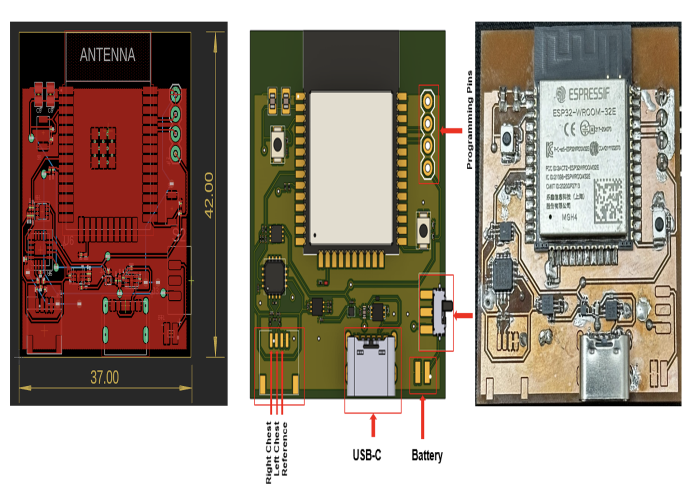


Fig. 8: (a) PCB Layout (b) 3D Model (c) Assembled Device

The waveform shown in Figure 10 is the output received from the device. This is a zoomed in view of 70 seconds of recorded data. This window is showing data from 40 to 50 seconds of recorded data. The ECG data is being sampled at 100 samples per second.

B. Mechanical Design

The mechanical design aimed to meet the specifications of both software and hardware components, particularly in the context of developing a stress indicator for drivers. This involved exploring diverse approaches to seamlessly integrate the device into an operational vehicle.

Addressing software requirements was pivotal, with a focus on Signal Fidelity Assurance. This required precise ECG signal transmission without distortion, accomplished through the implementation of noise-filtering algorithms. Real-Time Monitoring Compatibility was equally crucial, demanding seamless, delay-free monitoring enabled by efficient data processing for prompt stress level updates.

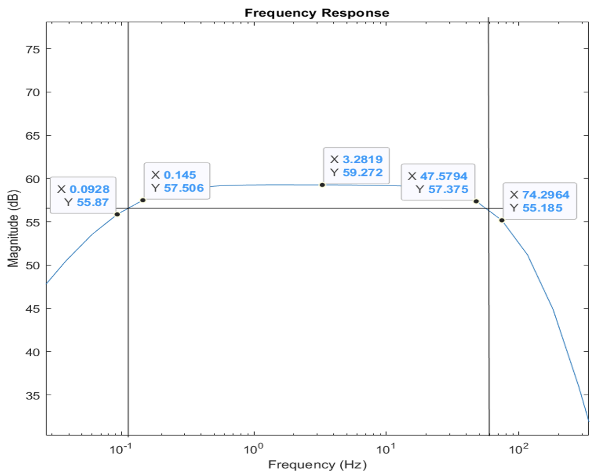


Fig. 9: Frequency Response

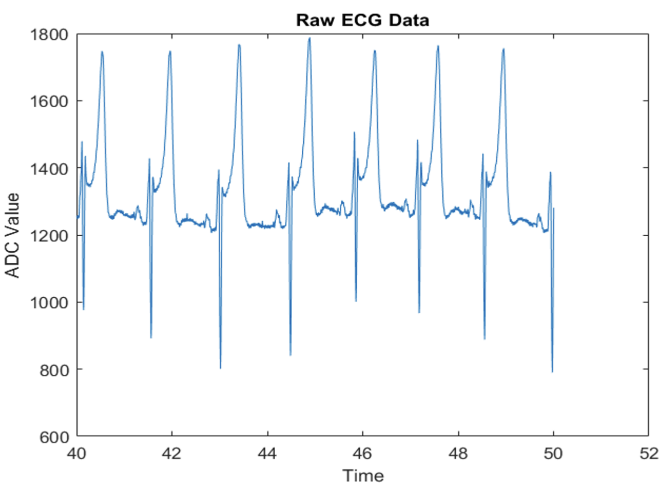


Fig. 10: ECG waveform output from the device

On the hardware front, meeting a comprehensive set of requirements was essential for successful stress monitoring integration. Full Part Inclusion Capability mandated seamless integration with vehicle components, ensuring compatibility with various car models. Effective Physical Enclosure demanded a robust, protective casing meeting safety standards and allowing easy maintenance access. Heat Resistance required integrating materials and cooling mechanisms to prevent overheating during prolonged operation, while Motion Restriction necessitated stabilization features to ensure accuracy during dynamic driving conditions.

The Seat Belt Module, despite offering advantages in stress monitoring such as enhanced signal quality and user compliance, presented challenges like potential movement along the seat belt and discomfort during extended use.

1) Objective 1: Mechanical Design:: The material selection and assembly process for the stress monitoring device underwent a meticulous reevaluation following the breakage of the initial Resin-printed snap joint, as shown in Figure 11 . In response, Polylactic Acid (PLA) emerged as the preferred material due to its superior durability and resilience against impact forces. The device was subsequently reprinted using PLA, addressing the shortcomings of the initial Resin print and enhancing overall durability, particularly in the critical snap joint. The assembly process encompassed the precise integration of a PLC board, customized slots for electrodes, battery, and switch assembly, as well as the strategic placement of a shock absorber and anti-slippery pad. These measures collectively contribute to the creation of a robust and stable stress monitoring solution, addressing challenges encountered in the initial Resin print and instilling confidence in its real-world effectiveness. Rigorous testing and validation will follow to ensure the seamless integration and success of these material and assembly enhancements. We performed a deformation analysis on the module, revealing a significantly improved result compared to the previous assessment. The former maximum structural deformation was recorded at 0.225 meters, while the new value indicates a substantially reduced deformation, measuring only 0.0004 meters, as illustrated in Figure 12.

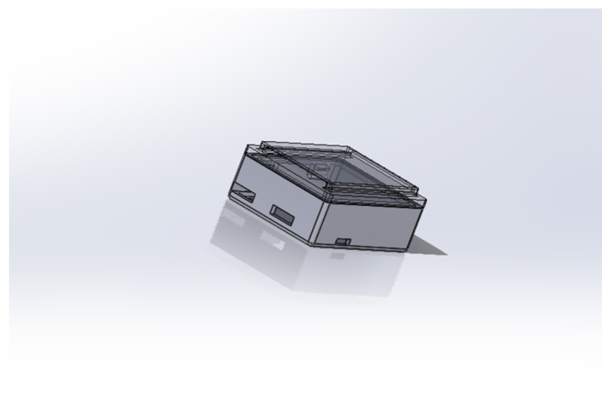


Fig. 11: Proposed enclosure design for accommodating the sensor

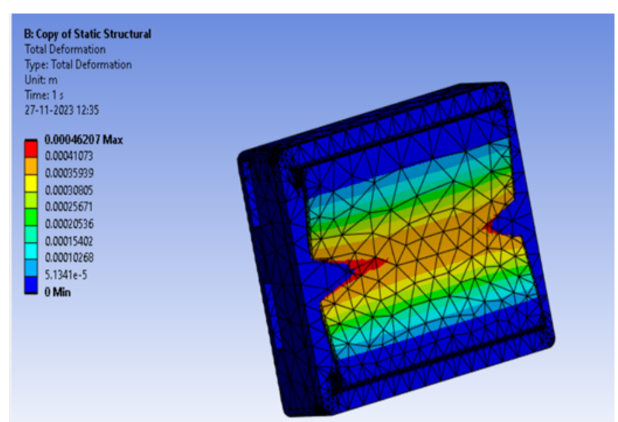


Fig. 12: Stress Analysis of proposed design

2) Objective 2: MATLAB Optimization of Seat Movement:: Precision Enhancement in Seat Movements utilized MATLAB algorithms to improve rotation and front-back motion precision. Arresting Module Motion along the Seat belt addressed challenges of excessive motion, enhancing module stability during various driving conditions.The MATLAB code establishes a linear regression model to predict seat belt lengths based on rotational angles and translation lengths. The data, organized in a table, includes information on individuals and corresponding measurements. The model is fitted using the 'fitlm' function, and predictions are made for a set of fixed rotational angle and translational length values. Additionally, 95 confidence intervals for the model coefficients are computed using the 'coefCI' function. This approach provides a statistical framework for estimating seat belt lengths in relation to specific geometric parameters, contributing valuable insights for potential applications in automotive safety. Insights gained from deformation analysis informed both mechanical design revisions and algorithm development in MATLAB to enhance seat movement precision and module stability. The predicted seat belt length was 30.21, with a 95% confidence interval of [-0.48, 0.83].

The sympathetic nervous system plays a key role in the body's "fight or flight" response, which is activated during stressful situations. By placing electrodes at the upper chest, the device captures electrical signals generated by the heart under the influence of sympathetic activity. This is shown in real-time in Figure 13. This enables the monitoring of physiological changes associated with stress, providing valuable insights into an individual's stress levels. The precise placement of electrodes on the upper chest ensures accurate and reliable data collection, contributing to the device's effectiveness in assessing and managing stress-related responses.

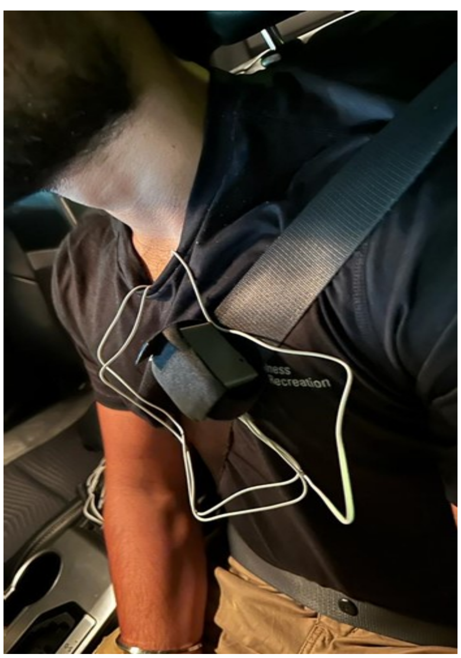


Fig. 13: Sensor placement in trial

VII. RESULTS

A. Mechanical Design

The mechanical design successfully met software and hardware specifications for a stress indicator. Software requirements focused on signal fidelity and real-time monitoring, while hardware addressed full part inclusion, physical enclosure, heat resistance, and motion restriction, overcoming challenges with the Seat Belt Module.Reevaluating material and assembly processes led to adopting Polylactic Acid (PLA), enhancing durability. The subsequent PLA printing and meticulous assembly resulted in a robust stress monitoring solution. Deformation analysis showed a significant improvement, reducing maximum structural deformation from 0.225 to 0.0004 meters.MATLAB optimization improved seat movement precision, addressing excessive motion and enhancing module stability during driving. Linear regression predicted a seat belt length of 30.21, with a 95% confidence interval of [-0.48, 0.83]. Insights from deformation analysis informed mechanical design and algorithm development for improved precision and stability.

B. Training Phase

Assessing the performance of machine learning models can be done with the standard classification metrics which are extensively used in previous literature. In this study, the metrics used are listed below along with their mathematical formulae.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$F1\ score = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

ML Algorithm	Accuracy	F1	Recall	Precision
K-NN	0.61	0.57	0.57	0.59
Decision Trees	0.68	0.67	0.68	0.68
Random Forest	0.72	0.70	0.69	0.73
XGBoost	0.65	0.59	0.58	0.65

TABLE II: Results of Classical ML Algorithms

These metrics were used to quantify the performance of ML models, the results of which are tabulated in Table II. In the analysis of classical machine learning models applied to ECG and heart rate data, superior performance was observed in the case of Random Forest, particularly in achieving a commendable recall rate—a crucial metric for minimizing false negatives. The class samples in the dataset were found to have imbalances, but despite attempts at oversampling, satisfactory results were not achieved. Consequently, the models were trained on the original dataset. Feature correlations with the output were analyzed, revealing that a strong correlation existed between mean HR and standard deviation in heart rate compared to other features. For feature importance scores, Random Forest was employed, and the top six features were selected for model training. The features chosen for training included the standard deviation of ECG and HR, skewness, and HRV-related features SDNN and RMSSD. Despite achieving relatively successful outcomes, the overall performance of classical ML models was deemed sub-optimal for practical deployment in healthcare settings [1].

In addition to standard classification metrics being employed for the assessment of DL model performance, the accuracy vs. epoch graph was also analyzed. The accuracy vs epoch graph for a deep learning model during training, typically exhibits an increasing trend initially, reflecting the model's learning process. This graph for the VGGNet architecture trained on the DriveDB dataset is presented in Figure 14. From the graph, it is observed that the accuracy of the model increases as the number of epochs increases. This is an indication that the model is learning from the training data and improving its ability to make accurate predictions.

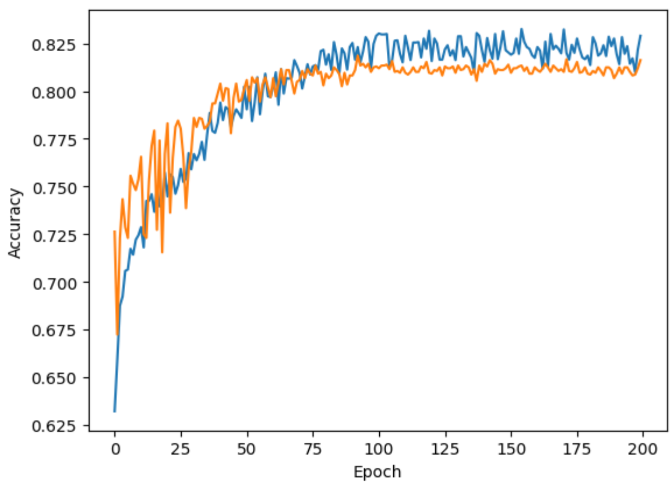


Fig. 14: Accuracy vs Epoch for VGGNet Architecture

Class	Accuracy
1	0.7497
3	0.3498
5	0.7727

TABLE III: Accuracy Scores for Multi-class Classification

From Table III, the multi-class classification DL model demonstrated varying accuracies across different classes, with an accuracy of **0.7497** for Class 1, **0.3498** for Class 3, and a notably higher accuracy of **0.7727** for Class 5. These scores provide insights into the model's effectiveness in distinguishing between the specified classes. Class 3 posed a greater challenge for detection as it represents moderate stress, a nuanced state that proves difficult for the model to discern. The difficulty arises from the resemblance of the ECG waveform in this class to the stressed ECG patterns observed in class 5, contributing to the lower accuracy score of 0.3498 for class 3.

C. *Testing Phase*

A Flask web application was developed and deployed on the Jetson Nano platform to facilitate real-time monitoring and analysis of electrocardiogram (ECG) signals. The primary objective was to create an interactive interface that provides users with a visual representation of live ECG data and predictions generated by a machine learning model. The Flask app incorporated WebSocket communication through the SocketIO library to establish a seamless and responsive connection between the server and the client.

The app's front end utilized Plotly, a web-based graphing library, to dynamically plot and update the live ECG signals in real time. The ECG data was transmitted to the server via a UDP socket, processed, and then forwarded to the client for visualization. Additionally, the pre-trained model from Section VII-B, was integrated into the application to perform real-time predictions based on the received ECG data. The predictions were relayed to the driver, allowing users to observe and analyze the model's assessments alongside the live ECG signals.

VIII. Conclusion

In summary, this study explored the viability of a real-time stress monitoring system for drivers, given the close correlation between stress and aggressive driving—an eminent contributor to road fatalities. The investigation delved into real-time ECG signal collection via sensors, culminating in the development of hardware schematics encompassing Power, Analog Front End, and WiFi modules. Proposals were outlined for the mechanical design of sensor placement on human subjects to optimize data acquisition. Additionally, diverse ECG signal processing techniques were scrutinized, resulting in a comprehensive pipeline architecture from data cleaning to feature extraction. The study thoroughly deliberated on the feasibility of Machine and Deep Learning techniques under diverse conditions, validated through extensive experimentation.Compared to the performance of classical ML approaches, deep learning's performance was the best in terms of various metrics. We used our trained deep learning model to quantify stress in real time as the model learned the essential patterns from the open source dataset to quantify stress based on the ECG signal and extracted features corresponding to different stress levels.Additionally to validate the performance of our system the future scope envisions to conduct surveys with participants while taking data to enhance the performance of our model.It's important to note that the current focus is on clinical experimentation, and the future scope envisions the study progressing towards broader applications and impact.

IX. Future Scope

The future scope of this study lies in the refinement of the proposed real-time stress monitoring system for drivers, with a focus on addressing areas like conducting rigorous testing of the prototype in simulated and real-world driving scenarios to evaluate its reliability and performance under diverse conditions.Alternative methods and materials for sensor placement on human subjects can be explored to enhance comfort and accuracy of data acquisition. Ergonomic considerations and user preferences for the placement of stress monitoring sensors can be investigated to encourage widespread acceptance and adoption.Possibilities for seamless integration of the stress monitoring system with existing vehicle safety systems can be investigated.Feasibility of continuous, long-term stress monitoring to identify patterns and trends in driver stress levels can be explored.By addressing these future considerations, the study can contribute significantly to the development of an effective, practical, and ethical real-time stress monitoring system for drivers, ultimately promoting safer road conditions and reducing the incidence of aggressive driving-related accidents.

REFERENCES

[1] OpenAI. (2021) Chatgpt. [Online]. Available: https://chat.openai.com

[2] J. Healey and R. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, 2005.

[3] J. L. Deffenbacher, E. R. Oetting, and R. S. Lynch, "Development of a driving anger scale," *Psychological Reports*, vol. 74, no. 1, pp. 83–91, 1994, pMID: 8153239. [Online]. Available: https://doi.org/10.2466/pr0.1994.74.1.83

[4] V. Balasubramanian and K. Adalarasu, "Emg-based analysis of change in muscle activity during simulated driving," *Journal of Bodywork and Movement Therapies*, vol. 11, no. 2, pp. 151–158, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1360859207000034

[5] M. P, S. S. J, J. P.K, R. R. Chandran, S. Krishnan, and S. Shiny, "Ecg based stress detection using machine learning," in *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2022, pp. 1–5.

[6] K. Tzevelekakis, Z. Stefanidi, and G. Margetis, "Real-time stress level feedback from raw ecg signals for personalised, context-aware applications using lightweight convolutional neural network architectures," *Sensors*, vol. 21, no. 23, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/23/7802

[7] P. Bobade and M. Vani, "Stress detection with machine learning and deep learning using multimodal physiological data," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 51–57.

[8] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing wesad, a multimodal dataset for wearable stress and affect detection," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ser. ICMI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 400–408. [Online]. Available: https://doi.org/10.1145/3242969.3242985

[9] J. Healey and R. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, 2005.

[10] F. Buendía-Fuentes, M. A. Arnau-Vives, A. Arnau-Vives, Y. Jiménez-Jiménez, J. Rueda-Soriano, E. Zorio-Grima, A. Osa-Sáez, L. V. Martínez-Dolz, L. Almenar-Bonet, and M. A. Palencia-Pérez, "High-Bandpass filters in electrocardiography: Source of error in the interpretation of the ST segment," *ISRN Cardiol*, vol. 2012, p. 706217, Jun. 2012.