Proceedings of the 9th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC-2025)
DVD Part Number: CFP25OSV-DVD; ISBN: 979-8-3315-7472-7

Embedded Machine Learning for Early Detection of Heart Attack Symptoms

Vivek M G
Computing Technologies,
SRM Institute of Science and Technology,
Kattankulathur, India
vm4512@srmist.edu.in

S Raghuram
Computing Technologies,
SRM Institute of Science and Technology,
Kattankulathur, India
rs0657@srmist.edu.in

Dr.R Jeya
Computing Technologies,
SRM Institute of Science and Technology,
Kattankulathur, India
jeyar@srmist.edu.in

Abstract— **Heart attacks are the most common cause of early deaths and accidents, and early signs are often missed. In this project, machine learning was incorporated into electric and fuel-powered vehicles to enable it to predict a heart attack when the driver is on the road driving the car. The system constantly measures physiological data such as heart rate m oxygen levels, and sweat, and processes it at the place with the help of a Neural Network Binary Classification model. Also, it examines signs of possible heart problems. It secures the life of the driver, as it gives immediate warnings to the driver, thus minimising the risks of accident due to cardiac conditions. The strategy helps in improving the safety of the vehicles and the quality of healthcare since vehicles become smarter and capable of responding to health emergencies.**

Keywords—Heart Attack Prediction, Embedded Machine Learning, Real-Time Health Monitoring, Driver Safety, Onboard Health Detection, Physiological Signal Analysis

I. INTRODUCTION

Heart attacks are a common cause of sudden death and accidents in the road, they often end without any prior warning. A lot of drivers have early symptoms which are not detected in time, which leads to life threatening conditions. According to the World Health Organization, cardiovascular diseases remain the leading cause of mortality worldwide, accounting for nearly one-third of all deaths, and their occurrence during driving poses a dual threat to both individual health and road safety. Although wearable devices and driver assistance systems exist, they often rely on internet connectivity, external monitoring, or are not designed for continuous, real-time use in vehicles, which leaves a critical gap in driver health monitoring.

Our project uses machine learning, which is embedded in cars, to predict the symptoms of a heart attack in real-time and, as a result, eliminates this problem. Physiological signals such as heart rate, blood pressure, oxygen levels, and sweating, are continuously monitored by the sensors in this system. This data is locally processed by a lightweight machine learning model that is capable of identifying the potential risk of a heart attack. Unlike traditional approaches, the novelty of this system lies in its on-device processing capability, independence from cloud servers, and its integration directly within vehicles, making it suitable for both electric and fuel-powered cars.

A driver can take precautionary steps or seek medical assistance by receiving an immediate warning from the system if a cardiac arrest pattern is detected. This initiative enhances driver safety, reduces accidents and enhances modern vehicles by integrating AI-driven health monitoring in both electric and fuel-powered vehicles. The objectives of this study are to design a system for continuous monitoring of vital signals in drivers, develop and deploy an embedded lightweight ML model for early detection of heart attack symptoms, and evaluate its role in enhancing road safety and advancing health-conscious intelligent transportation systems.

II. LITERATURE SURVEY

Machine learning algorithms and a real-time cardiovascular system help to identify heart attacks with help of algorithms with Support Vector Machine, Naïve Bayes, Artificial Neural Networks, Random Forest, and Logistic Regression, SVM demonstrating the highest accuracy. The study combines two open-access datasets (Cleveland and Statlog Heart Disease) to improve prediction robustness. An Arduino-based real-time monitoring system records heart rate, temperature, and humidity, transmitting data to a cloud server for mobile app-based patient-doctor communication. Despite its advantages, the system does not include yawning detection, lacks an alert mechanism for unsafe driving behavior, and relies on continuous sensor accuracy, which may be impacted by environmental factors [1].

Various hardware components and algorithms are integrated to provide real-time monitoring for utilizing IoT and machine learning for safety support and accident tolerance. The system detects driver fatigue by deploying Raspberry Pi, cameras, alcohol sensors, gas sensors, and machine learning algorithms, such as the Eye Aspect Ratio and Mouth Aspect Ratio. Yet it is devoid of a yawning detection mechanism, an alert system for notifying vehicle owners about change in behaviour detected by alcohol sensors, and is heavily reliant on sensor accuracy, which may be impacted by poor illumination or sensor errors [2].

This study uses a public health dataset that was derived from the Cleveland, Hungary, Switzerland, and Long Beach V databases. It selects 14 vital features from 76 attributes. Preprocessing techniques, including the Lasso algorithm for feature selection and the Isolation Forest method for outlier management, were implemented. Numerous machine learning algorithms such as Logistic Regression, K-Nearest Neighbours, Decision Trees, Random Forest, SVM, and XGBoost were evaluated, and dimensionality reduction techniques increased the efficacy of the model. Furthermore, a sequential neural network employing three dense layers, ReLU activation for hidden layers, Sigmoid activation for output and Dropout layers to prevent overfitting was employed to implement a deep learning model. Nevertheless, the study is constrained by the absence of clinically approved PPG-based blood pressure sensors, which restricts medical accuracy. Despite the proposal of a mobile application, real-world testing was not conducted, and the model's generalisation remains unvalidated for broader populations. Additionally, regions with inadequate network connectivity may encounter complications as a consequence of cloud connectivity [3].

In order to enhance the efficacy of the model and reduce its dimensionality, this study adopted an embedded feature selection technique that implemented LinearSVC with L1 norm. Deep Neural Network (DNN) was developed to further improve classification based on the use of particular features to forecast cardiac diseases. The IQR method was used to remove the outliers and the data standardisation was used to ensure feature consistency during the preprocessing stage. The initialiser was suggested after considering the various weight initialisation methods, and it was concluded that the initialiser was the most effective in avoiding gradient problems and weight stability. The study has however been withdrawn owing to the issue of the publication and peer-review process which might have impaired its reliability. Also, it is not clear how the model is applicable to larger datasets, although the Kaggle dataset was used. Further optimization with the depth and parameter adjustments of the DNN, was proposed but it was not carefully studied. It could have impacted the overall performance of the model, which was not evaluated in terms of the effect of the removal of the features properly [4].

The study postulates a smart healthcare framework (SHDML) to detect heart disease in real time, in which machine learning and deep learning models are used in association with photoplethysmogram(PPG) sensors connected to an ATmega32 microcontroller for heart rate monitoring. The study utilizes data from the Framingham Heart Study and PPG sensors that measure heart rate and are connected to an ATmega32 microcontroller. The experiment relies on the results of the Framingham Heart Study and PPG sensors and is processed with the help of normalisation and the ability to overcome missing values. Models such as SVM, Logistic Regression and ANN methods as well as deep learning models like CNNs were employed to improve the accuracy in the prediction. Storage and processing of data was done through Firebase cloud and results presented on Android and desktop apps. Limitations, however, are the size of the ATmega32-based device, which is too huge to be used in real life, and the lack of further health monitoring sensors, like temperature and blood pressure sensors. The user interface and the quality of model prediction should be improved further. In addition, the variety of datasets available to validate the model is limited, and they are not large enough to ensure the goodness of the model in generalization [5].

This article offers an Internet of Things (IoT) accident prevention system based on machine learning models such as Random Forest, LightGBM, and XGBoost. This paper relies on the Countrywide Traffic Accident Dataset, a dataset of 2.8 million U.S.-based car accidents between 2016 and 2021. In order to increase the accuracy of the accident predictor, the IoT devices collect data on the location, time, weather, wind speed, visibility and temperature. The performance of the model was tested using F1-score, ROC-AUC, precision, recall, and accuracy. The model, however, has limited applicability to other regions due to the fact that the dataset used is on the United States that may give rise to bias. The performance of the model in the real world might also be different in relation to some of the datasets or real-time conditions [6].

This system integrates smart wearable devices, smartphones, in-vehicle data collection devices, and intelligent transportation information collection devices for driver state monitoring. Machine learning algorithms analyze driver behavior and detect signs of fatigue. However, the accuracy of the monitoring system may be affected by external factors and inherent limitations of machine learning algorithms[7].

This study implements IoT sensors such as alcohol and air pressure sensors for sobriety checks and uses machine learning to detect micro-sleep and frequent yawns, indicating drowsiness. A Raspberry Pi 3B+ microprocessor processes data from sensors and cameras, performing real-time analysis of fatigue indicators. However, challenges include sensor calibration requirements, the need for valid breath samples, and the limited computational resources of the Raspberry Pi, which may affect system scalability [8].

This system implements an Arduino microcontroller with incorporated sensors, such as a webcam, accelerometer, infrared, and MQ-3, to avoid and detect accidents. A real-time data monitoring threshold-based alert system, V2V communication for safe distance maintenance, RFID technology for license verification, alcohol sensors to prevent vehicle ignition, and accelerometers to detect accidents and send alerts via GPS and GSM are some of the features. Additional safety measures consist of an automatic braking system that is activated by red traffic signals and Eye Blink Monitoring (EBM) that employs infrared sensors. However, the capacity of the system are contingent upon the functionality of the sensors and the environmental conditions. RFID registration constraints, high integration costs, data privacy concerns, and implementation complexity are among the limitations [9].

This study employs ML classifiers such as SVM, Logistic Regression, ANN, KNN, Naïve Bayes, and Decision Trees for heart disease diagnosis. Feature selection techniques, including Relief, MRMR, LASSO, and the novel FCMIM algorithm, enhance model performance. The Cleveland Heart Disease dataset (297 samples, 13 attributes) was used with Leave-One-Subject-Out Cross Validation (LOSO CV) for evaluation. However, the limited dataset size constrained training complex models like deep neural networks. The study also focused on specific ML classifiers without exploring ensemble or hybrid models. Further optimization is needed for generalization across different datasets and real-time applications in resource-limited environments. Moreover, the lack of diverse and larger datasets for validation restricts the model's generalizability [10].

III. COMPARATIVE STUDY AND ANALYSIS

The proposed research is unique in that it uses a balanced, synthetic dataset that includes physiological parameters such as heart rate, body temperature, SpO_2, and GSR (sweat levels). In contrast to many previous works that rely on pre-existing datasets or focus on isolated parameters, our system integrates multiple sensors and uses an embedded neural network on the Arduino Nano 33 BLE Sense Rev2 to perform real-time classification completely offline. As a result, it is ideal for situations involving cars where network access is limited. Training with the lightweight TensorFlow Lite architecture allows the model to operate efficiently in low-power scenarios and achieve high accuracy.

In contrast, Paper 1 makes use of the Cleveland and Statlog datasets along with traditional machine learning models such as SVM and ANN. Its lack of yawning detection and alert systems and dependence on cloud infrastructure restrict its offline usability. Despite focussing on EAR and MAR fatigue detection and utilising a Raspberry Pi, Paper 2 fails to alert users to the presence of alcohol or risky behaviour. Paper 3 explores deep neural networks using public datasets and advanced preprocessing, but it lacks clinical validation and real-world testing, especially when conducted offline. While Paper 4 makes use of DNN and embedded feature selection, the study was withdrawn, and its generalisability is in doubt.

Despite lacking other necessary sensors and relying on a sizable microcontroller, Paper 5 offers a clever framework that makes use of PPG sensors and Firebase cloud services. Using environmental data, Paper 6 develops an accident prevention system, albeit one that is geographically restricted and not embedded. Paper 7 uses a hybrid IoT setup to monitor driver behaviour, but it is heavily reliant on external factors. Although Paper 8 is computationally constrained and difficult to scale, it detects microsleep using the camera and alcohol sensors of a Raspberry Pi. Despite implementing an advanced alert system using a combination of sensors and actuators, Paper 9 is costly, necessitates complex integration, and presents privacy concerns. Finally, in Paper 10, various classifiers are tested using the Cleveland dataset; however, the dataset lacks embedded deployment and is too small for deep learning models.

In contrast, our project demonstrates that a small, real-time health monitoring system on an embedded platform can be developed using lightweight models. There is potential for improvement, but synthetic data limits clinical generalisability. Future studies could include EAR/MAR for fatigue detection and GSM/GPS for alerting, maintaining computational efficiency while closely resembling the real-world safety systems discussed in the reviewed papers. Table 1 below summarizes the limitations of prior studies and the corresponding improvements proposed in this work.

TABLE 1: LIMITATIONS OF PRIOR STUDIES AND PROPOSED IMPROVEMENTS

Ref	Limitations & Proposed Improvements
[1]	No alerts/fatigue detection, sensor-dependent; Offline alerts with multi-parameter input
[2]	No alcohol alerts, lighting-sensitive ; Physiological + stress signals, robust operation
[3]	No real-world or embedded validation ; Real-time low-power embedded deployment
[4]	Retracted, dataset limits ; Hardware-tested, stable model with efficient features
[5]	Bulky, missing sensors, cloud-reliant ; Compact, multi-sensor, low-power embedded design
[6]	Accident-specific, geographic bias ; Broader health-focused dataset
[7]	Accuracy drops in environment ; Physiological signals for reliability
[8]	Calibration issues, limited compute ; Optimized low-power embedded inference
[9]	High cost, privacy risks, complex setup ; Lightweight, privacy-preserving model
[10]	Small dataset, no deep learning ; Large dataset + deep learning on embedded hardware

Proposed work	Needs clinical validation ; 96.64% accuracy, offline alerts, multi-sensor low-power design

IV. Hardware Components and Sensors

In this section, we detail the hardware components and sensors integrated into our embedded system for the early detection of heart attack symptoms. Each component's functionality, specifications, and role within the system are discussed to elucidate their contributions to accurate and reliable physiological monitoring.

1. Arduino Nano 33 BLE Sense Rev2

Our system is centred on an Arduino Nano 33 BLE Sense Rev2, which is a microcontroller unit that coordinates the activities of data acquisition, processing and communication. It is designed with the nRF52840 microcontroller that uses 32-bit ARM cortex-M4 processor with low power consumption of 64Mhz, and it balances processing speed and power consumption. Board has 1MB flash storage and 256KB SRAM, which is enough to store machine learning models and handle sensor data effectively. In the case of wireless communication, it offers Bluetooth 5.0 Low Energy (BLE) which allows the transmission of real time data to external devices easily in order to monitor and analyse them. Along with that, the board incorporates various onboard sensors such as a 9-axis inertial measurement unit (IMU), temperature, humidity and pressure sensors, and a digital microphone among others, which increase its ability to record a broad contextual and environmental spectrum of data. The Figure 1 below is the Arduino Nano 33 BLE Sense rev2 pin diagram, which depicts the layout and the location of inputs/output pin.

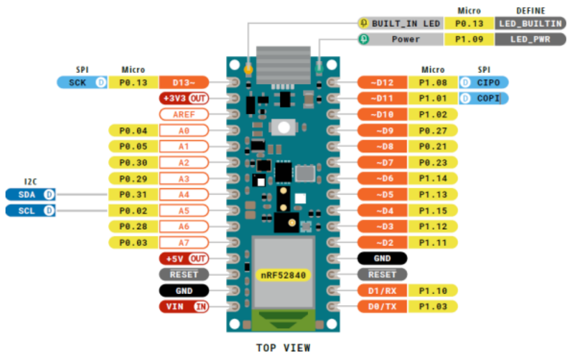

Figure 1 - Arduino Nano 33 BLE Sense Rev2 Pin Diagram

2. MAX30102 Pulse Oximeter and Heart-Rate Sensor

The MAX30102 Pulse Oximeter and Heart Rate Sensor is an essential component for non-invasive monitoring of critical health parameters, specifically blood oxygen levels (SpO_2) and heart rate. These measurements are vital for assessing cardiovascular health and identifying potential anomalies at an early stage. The sensor integrates red and infrared LEDs, a photodetector, optical components, and low-noise electronics that minimize interference from ambient light. Ensuring highly accurate readings. It operates with 1.8V supply for the sensor core and a separate 3.3V supply for the LEDs, optimizing power consumption for wearable and embedded application

Communication with the microcontroller is achieved through the I²C interface, enabling efficient and reliable data transfer. The MAX30102 functions on the principle of photoplethysmography, which measures variations in blood volume within tissue by detecting changes in light absorption. Red and infrared LEDs emit light into the skin, and the photodetector measures the amount of light either reflected or transmitted. Because oxygenated and deoxygenated haemoglobin absorb light differently at specific wavelengths, the ratio of red to infrared light absorption can be used to calculate blood oxygen saturation. Heart rate is determined from the periodic fluctuations in light absorption corresponding to heartbeats.

With its integrated optical components, ambient light cancellation, low-noise circuitry, dual-voltage operation, and I²C communication, the MAX30102 provides accurate, real-time monitoring of vital signs, making it highly suitable for integration in wearable health devices, fitness trackers, smartwatches, and other embedded health monitoring systems. Figure 2, shown below, illustrates the internal working principle of the MAX30102, highlighting the photodetector along with the red and infrared LEDs.

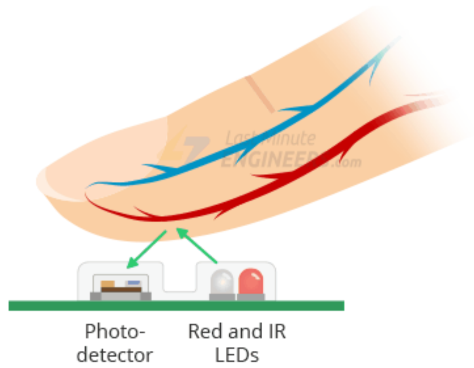

Figure 2 - MAX30102 Pulse Oximeter and Heart-Rate Sensor

3. Galvanic Skin Response (GSR) Sensor

The Galvanic Skin Response (GSR) sensor measures the electrical conductance of the skin, which varies with sweat gland activity—a physiological response closely linked to stress and emotional arousal. These factors can influence cardiac events, making the sensor particularly useful for monitoring heart health under varying physical or mental conditions.

The GSR sensor can detect variation in the skin conductance and thus gives information on the activity of the autonomic nervous system and can be used to gauge levels of stress. It connects to the microcontroller through the analog input pins, and it is thus capable of performing real-time continuous monitoring of physiological responses.

Sensor is used to measure the current resulting when a small constant voltage is applied to two electrodes on skin. When skin resistance is decreased by high activity of sweat glands, voltage signal is produced similar to that of skin conductance. This analogue signal captures stress in changes and emotional arousal which is useful to early detect cardiac risks. The sensor of GSR that would be used in this work is represented in figure 3 below.

Figure 3 – Galvanic Skin Response Sensor

4. TMP117 Temperature Sensor

TMP117 is a digital temperature sensor of high precision that is essential in measuring body temperature changes, which may indicate physiological stress or infection of cardiac activity.

It utilises silicon based band gap sensing element to operate the sensor. The voltage through this element is varied predictably with the body temperature. Such voltage alterations are sampled by 16-bit internal analog-to-digital converter (ADC), enabling highly precise temperature readings with a resolution of 0.0078°C and an accuracy of ±0.1°C.

The interface to the microcontroller is the I^2C interface, which makes it easily integrated with the Arduino Nano 33 BLE Sense Rev2. This provides the ease of data collection and real-time tracking, which is indispensable to embedded health monitoring gadgets and wearable devices.

The TMP117 is a high-reliability, low-power consumption, and highly accurate measurement device that can be used in continuous health monitoring in wearable trackers, medical devices, and environmental monitoring devices. Proper temperature measurements aid in the detection of fever, physiological stress, or inflammation, which can be the indirect indication of cardiac risk. The temperature sensor used in this study is TMP117 as illustrated in Figure 4 below.

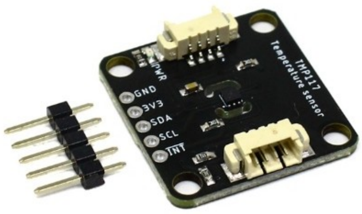

Figure 4 – TMP117 Temperature Sensor

V. Dataset Description

Embedded machine learning was used to identify the symptoms of a heart attack using a synthetic and balanced dataset. This data set was also customised to represent real-world physiological measurements that are applicable and practical to measure with the use of the Arduino Nano 33 BLE Sense Rev2 board. The data can be used to train the supervised learning and is designed to help with a binary classification that determines whether a patient has a risk of a heart attack or not.

A. Dataset Overview

The dataset that was used in this project comprises 50,000 samples each of which is a specific physiological health parameter that is relevant in early detection of heart attacks. It is artificial yet has a high similarity to a real-world medical situation, thus ensuring that there are equal positive (high risk) and negative (low risk) cases necessary to classify in a binary manner.

B. Features of the Dataset

Dataset comprises following **9** key attributes:

Physiological Parameters

The data set has four physiological parameters as input features. One of the primary indicators of cardiovascular health is Heart Rate (HR) which is measured in beats per minute (BPM). The sensor is the MAX30102 that measures blood oxygen saturation (SpO_2) in a percentage of the arterial blood as an indicator of oxygen availability. The TMP117 sensor gives body temperature (in degrees Celsius), which gives data about the metabolic activity and stress responses. Lastly, GSR that is expressed in microsiemens (µS) is utilised in recording modifications in skin conductance related to levels of stress and emotional arousal.

Target Variable

Heart Attack Risk is Dataset target variable, that is being binary classification highlighting presence (1) or absence (0) of heart attack risk.

C. Data Preprocessing

Data pre-processing was done, before mode development. The continuous variables (Heart Rate, SpO_2, GSR and Temperature) were scaled with Min-Max scaling to ensure the similarity of input range. The binary value was named Heart_Attack_Prediction Attribute. To perform performance evaluation, the dataset was separated into performance evaluation sections of: training (70%), Validation (15%) and test (15%). Though it has been synthetically generated, the data were shaped in such a way that they do not contain extreme values that would skew model training.

D. Dataset Suitability for Embedded Deployment

The data is well versed to be used in embedded applications as it has a small feature set, comprising of four sensor-based measurements; Heart rate, SpO_2, Temperature and Galvanic Skin Response (GSR). The small feature space size means that a small input vector is produced making fast inference time, which is an important aspect in real-time systems. Moreover, these parameters are measurable through low-cost biosensors that are compatible with widely used microcontroller platforms, such as the Arduino Nano 33 BLE sense Rev2, thereby ensuring ease of integration in embedded systems. The dataset design also incorporates a simple feature structure with a binary output, which minimizes memory consumption and reduces computational overhead. Consequently, this leads to a low memory footprint and supports efficient model execution on edge devices.

E. Justification for Synthetic Dataset Use

A synthetic dataset was generated owing to the absence of publicly available datasets specifically designed for real-time, embedded heart health monitoring applications. The use of synthetic data is justified on several grounds. First, the dataset simulates medically plausible patterns and sensor readings consistent with real-life cardiac scenarios, thereby ensuring realism. Second, it gives complete control over class balance, which means that the presence of positive and negative samples is equal to avoid bias during training. Third, the method does not involve any privacy risks, no personal or patient-identifiable information is involved, and that complicates the ethical aspect. Lastly, synthetic data is useful in fast prototyping because it can develop faster iteration and testing at the development stage before the real sensor input is integrated.

F. Dataset Analysis: Correlation and Outlier Detection

To obtain information about the internal structure and quality of the physiological data utilised in this research, two exploratory methods of data analysis were used: a correlation heatmap and outliers detection using boxplot. These visualisations allowed seeing the statistical relationships between features and detecting the abnormalities that may impact the work of the machine learning model. The correlation heatmap showed how linear relationships existed between numerical features and the target variable and how strong they were, which informed the selection of features and confirmed their relevance in prediction. In the meantime, the boxplot allowed identifying outliers and extreme values per feature that might indicate edge cases or hardware-related noise in measurements that might affect the training of the model. Combined with each other, these approaches resulted in a comprehensive assessment of the quality of the data and informed preprocessing choices (normalisation, outlier treatment, feature selection, etc.).

1. Correlation Heatmap:

All numerical features were then compared to each other using the Pearson correlation coefficient to produce a correlation heatmap offering a visual representation of linear correlations between pairs of features. This visualisation may be used to determine possible redundancy or co-dependency within the dataset and is thus used to gain a clearer insight into how the features are related to each other and to the target variable. The following heatmap of this correlation, based on all physiological features in heart attack prediction, is presented in figure 5 below.

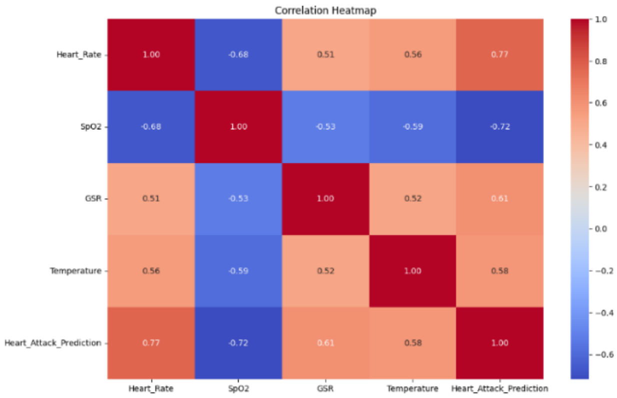

Figure 5 – Correlation Heatmap

There are various relationships that can be seen in the heatmap. There is a positive relationship between the rate of heart and the outcome of prediction of a heart attack and this positive relationship is strong, that is, high pulse rates are tightly connected with a high probability of heart attack. On the other hand, SpO 2 is negatively correlated to the target variable implying that decreased blood oxygen levels can be an important predictor of heart attack occurrence. Temperature and GSR indicate moderate positive correlations with the target, which means that the two features have complementary information on physiological abnormalities and stress responses.

Furthermore, the correlation of features, such as GSR and temperature or heart rate and temperature, prove that the interactions between physiological signals are possible and may influence each other in a synergistic way.

Besides that, comparison of features between two or more features, like GSR vs. temperature or heart rate vs. temperature, indicate the possibility of an interaction between physiological cues and their overall effect on heart well-being. Overall, the correlation heatmap validates the predictive relevance of the selected features and supports their inclusion in the embedded machine learning pipeline, ensuring a robust foundation for accurate classification.

2. Boxplot for Outlier Detection

A boxplot was employed to assess the distribution of features and to identify potential outliers, which can indicate meaningful edge cases in health measurements or signal hardware noise. Figure 6, shown below, presents this visualization for the primary physiological features in the dataset, including Heart Rate, SpO_2, GSR, Temperature, and the binary target variable, Heart Attack Prediction. The boxplot provides a comprehensive statistical summary of each feature, highlighting central tendency, variability, and the presence of outliers.

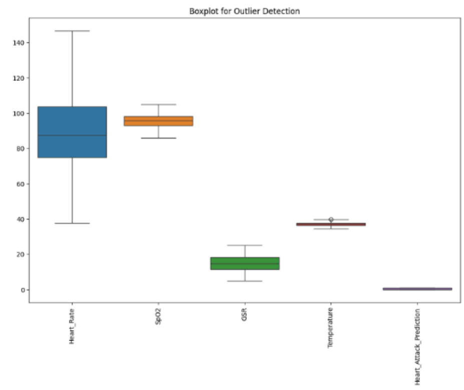

Figure 6 - Boxplot for Outlier Detection across Physiological Features

The heart rate feature analysis indicates the existence of the outliers on the high side and the interquartile range is high and this may be explained by high heart rate or stress cases in some individuals. The values of SpO_2 on the other hand, are highly concentrated on the upper side of the values; this is an indication of a fairly steady state of oxygen saturation in the subjects. There is moderate dispersion of GSR values with random deviations of extreme ends which comprise physiological responses to stress in few respondents. Temperature values are relatively homogeneous and values that are very small with minimal deviations indicate the presence of steady conditions of measurement.

The Heart Attack Prediction is a binary predictor and, hence, at the right place, two discrete values are presented, indicating that the number of classes is equal in the data. This visualisation plays a major role in data quality verification that may further inform preprocessing activities that includes normalisation and outlier management, which have played a critical role in ensuring faithful model training and correct predictive performance.

VI. Model Training and Evaluation

In this project, a supervised classification method is applied in which a neural network is used to identify the early warning signs of heart attacks on the basis of physiological data, namely the Heart Rate, SpO_2, Galvanic Skin Response (GSR) and Body Temperature. The neural network was chosen due to its ability to address non-linear relationships that are complex and at the same time be computationally efficient. The model was also trained on a synthetically generated and balanced dataset and then optimised to run in real-time by converting it to TensorFlow Lite (TFLite) format. This ensures compatibility with embedded hardware platforms such as the Arduino Nano 33 BLE Sense Rev2, where low latency and minimal memory usage are critical.

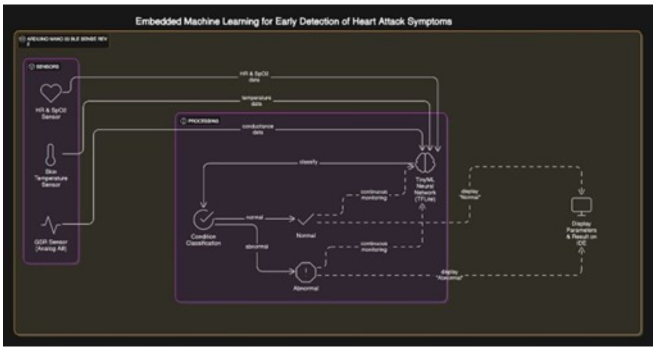

Figure 7 – System Architecture

The architecture of the system is shown in figure 7. The Arduino Nano 33 BLE Sense Rev2 provides a combination of real-time processing and several biomedical sensors that create an embedded machine learning system. The sensor measures physiological data, i.e., heart rate and SpO 2 values with the MAX30102 sensor, body temperature with the TMP117 sensor, and stress-related sweat activity with the GSR sensor. A lightweight TensorFlow Lite model deployed on the microcontroller processes these continuous inputs and classifies the driver's condition as normal or abnormal. Health condition is sent to a display unit and a buzzer is set to go off when there are indications of cardiac distress patterns. This architecture allows offline and low-latency inference, which is appropriate in car health monitoring, both in urban and rural areas.

A. *Model Architecture*

The neural network model that will be used in the research is a fully connected, feedforward network that is capable of classifying physiological input, such as Heart Rate, SpO_2, Galvanic Skin Response (GSR) and Body Temperature, to early detect heart attack symptoms in drivers. It is a lightweight model that was chosen based on its capacity to represent non-linear complex relationships among input features, and is computationally efficient enough to be deployed to its embedded systems. The network is structured with two dense hidden layers, each utilizing ReLU activation functions to learn non-linear decision boundaries, and dropout regularization is applied to mitigate overfitting under real-world, noisy sensor conditions. All the input features are normalised to give equal contribution in the training process and the output layer has a single neuron with a sigmoid activation function that gives binary classification: 0 no early symptoms were found and 1 possible early heart attack symptoms.

The fundamental supervised learning algorithm used in training the network is backpropagation which enables the network to differentiate between normal and abnormal physiological readings. The algorithm works with minimizing the loss between the predicted outputs and ground-truth labels until they are minimal. A forward pass is completed during training and calculates the predicted outputs and a backward pass corrects the error sent through the network to modify internal parameters, such as weights and biases. The process facilitates the network to maximise the mapping of physiological inputs to health-risk labels to attain precise classification. Backpropagation algorithm thus plays a central role in allowing the neural network to learn about the training data as well as making effective generalisations to unknown input.

Figure 8, placed below illustrates the backpropagation process through the hidden and output layers, highlighting how error signals are propagated backward and used to refine the model during training. By integrating continuous updates of network parameters with real-time physiological input, this architecture provides the foundation for deploying a reliable, embedded machine learning system on platforms such as the Arduino Nano 33 BLE Sense Rev2.

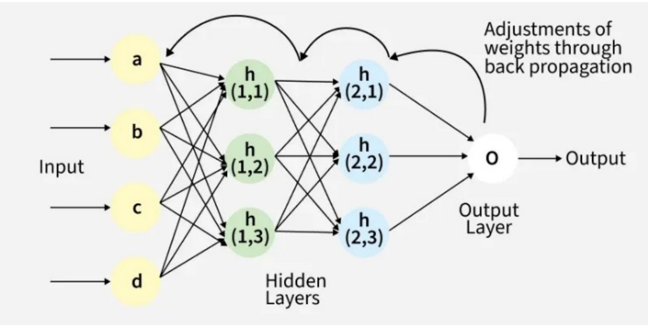

Figure 8 - Visualization of Backpropagation Through Hidden and Output Layers

B. Mathematical Formulation:

The neural network is trained using binary cross-entropy as the loss function, which is defined as:

$$L(y, \hat{y}) = -[ylog(\hat{y}) + (1 - y)log(1 - \hat{y})]$$

where y is the true label (0 = normal, 1 = abnormal), and $\hat{y}$ is the predicted probability. Gradient computation is performed to update the network parameters. The error at the output layer is calculated as

$$\boldsymbol{\delta(2) = \hat{y} - y}$$

and the gradient with respect to the output weights is given by

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}^{(2)}} = \boldsymbol{\delta}^{(2)} \cdot (\boldsymbol{a}^{(1)})^{\boldsymbol{T}}$$

This error is then propagated to the hidden layer, where the backpropagated error is calculated as

$$\boldsymbol{\delta}^{(1)} = (\boldsymbol{W}^{(2)})^{\boldsymbol{T}} \boldsymbol{\delta}^{(2)} \cdot \boldsymbol{f}'(\boldsymbol{z}^{(1)})$$

and the gradient with respect to the hidden-layer weights is

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}^{(1)}} = \boldsymbol{\delta}^{(1)} . \boldsymbol{x}^{\boldsymbol{T}}$$

Weights are updated iteratively using gradient descent according to the rule

$$\boldsymbol{W}^{(l)} := \boldsymbol{W}^{(l)} - \boldsymbol{\eta} \cdot \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}^{(1)}}$$

where η is the learning rate. This iterative update is applied across multiple epochs until convergence, allowing the network to minimize loss and learn the optimal mapping between physiological inputs and health risk labels. After training and validation, the model achieved an accuracy of 96.64%. Subsequently, it was converted to TensorFlow Lite and deployed on the Arduino Nano 33 BLE Sense Rev2. At runtime, forward inference is only done, and therefore real-time classification can still be achieved without the cost of backpropagation.

The four features that the neural network input layer takes are Heart Rate, SpO_2, Galvanic Skin Response (GSR), and Body Temperature. All features are normalised to have the same contribution in the training process that enhances convergence and model stability.

The network has two dense hidden layers, with all of them using ReLU activation function to gain knowledge on non-linear decision boundaries. Regularisation of dropout is implemented after every hidden layer in order to reduce overfitting and in this case especially where the model is implemented in real life with noisy sensor data.

The output layer is a neuron with a single output which is a binary classification and has a sigmoid activation function. In this case, 0 is used to depict the absence of early symptoms and 1 is used to depict possible early symptoms of heart attack. This configuration allows the network to translate physiological inputs into meaningful health risk predictions, making it suitable for embedded deployment on platforms such as the Arduino Nano 33 BLE Sense Rev2 for real-time monitoring.

C. Model Training Process

The neural network was also trained on the Adam optimizer because it has adaptable learning rates to apply throughout the training process making it converge faster and more steadily. The loss function applied in this case was binary cross-entropy loss, as it is most appropriate to use in a problem with two different classes, which in this example are high-risk and low-risk. The model was trained in multiple epochs and optimised batch-wise to minimise the use of memory and early stopping was based on the observed validation loss so as to avoid overfitting and achieve improved generalisation. A certain part of the dataset was reserved as a validation set to continuously assess the model with the unknown data during the training to get a consistent measure of the predictive performance of the model.

D. Evaluation and Performance Metrics

The model was tested on unknown data to determine its efficiency after training. The essential performance indicators were accuracy, which is the ratio of accurate predictions in total; precision, which is the count of correct predictions of the high-risk type; recall, which is the proportion of the actual high-risk cases that were correctly identified; and the F1-score, which gives a balanced score of the two indicators: precision and recall. These measures put together provide a holistic evaluation of the predictive accuracy of the model, especially in the area of the ability to detect possible symptoms of cardiac attack in its early stages correctly.

E. Embedded Deployment Readiness

For real-time deployment on microcontroller boards such as the Arduino Nano 33 BLE Sense Rev2, trained neural network model had been converted into TFLite format. TensorFlow Lite is an open-source, low-power deep learning framework developed by Google that enables on-device inference on mobile, embedded, and edge devices with constrained processing power, thereby reducing the need for cloud-based computation. Standard TensorFlow models are converted into a smaller, optimized TFLite format using techniques such as memory planning, operator fusion, and quantization, which decrease model size and improve inference speed, making them suitable for real-time, low-power applications.

TensorFlow Lite for Microcontrollers (TFLM) allows microcontroller-based systems to run a subset of the TFLite runtime without requiring an operating system. This enables deployment on devices with as little as 16 KB of RAM, including the Arduino Nano 33 BLE Sense, STM32, and ESP32. Post-training quantization was applied to reduce model size further and improve inference speed, while the inference logic was optimized to process real-time physiological inputs from onboard sensors, including body temperature, SpO_2, GSR, and heart rate.

TensorFlow Lite for Microcontrollers (TFLM) has made it by deploying the model in this manner, physiological data can be processed offline, providing low-latency, real-time classification for embedded health monitoring applications in automobiles. This approach has the potential to enhance early intervention for cardiac events. Future work will focus on incorporating additional physiological parameters and further development of the system into a fully integrated product.

VII. RESULTS AND DISCUSSIONS

The Performance of a balanced dataset of physiological parameters of Heart rate, SpO_2, Sweat levels and body temperature was recorded to be evaluated. The model training and optimization after being pre-processed and trained on a lightweight neural network was deployed after being converted into a TensorFlow Lite format, and executed on the Arduino Nano 33 BLE Sense Rev2. Standard metrics of classification including Accuracy, Precision, Recall, and F1-Score were used to evaluate the model. The performance of deployed model is summarised in Table II:

TABLE II: PERFORMANCE METRICS OF THE DEPLOYED MODEL

METRIC	**VALUE**
Accuracy	96.64%
Precision	95.12%
Recall	94.78%
F1-Score	94.95%

The results show that the model performs reliably even in resource-constrained embedded environments, achieving 96.64% accuracy. It has a good F1-score, which means that it has a good balance between precision and recall, which is essential in health monitoring to reduce false negatives. These results prove that the real-time heart attack detection with machine learning on microcontrollers is efficient and possible.

To evaluate the proposed lightweight neural network trained on Heart Rate, SpO_2, Body Temperature, and GSR, training and validation performance was monitored over 50 epochs using loss and accuracy graphs to assess convergence and generalization.

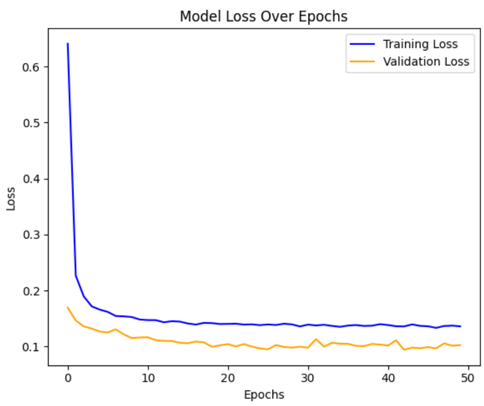

Figure 9 – Model Loss Over Epochs

Figure 9 shows the loss of training reduced rapidly in the early stages of the epochs, which means that the model was able to learn the reduction of classification error very fast. The validation loss also decreased and leveled off with a smaller value compared to the training loss indicating the model is not overfitting and can generalize well to unseen data.

The accuracy curves of the two datasets are shown in figure 10 below. The model also had fast increases on both training and validation accuracy on the first 10-15 epoch but the validation accuracy was always higher than the training accuracy. The final validation percentage of the model was about 96.64, which justifies its appropriateness in real-time use on the embedded hardware.

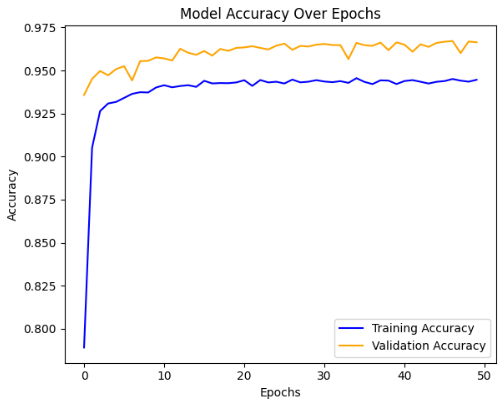

Figure 10 – Model Accuracy Over Epochs

Such plots validate that the model effectively learns significant patterns of the physiological inputs and is stable to training, which is desirable in health-critical areas like heart attack symptoms in drivers, among others.

VIII. CONCLUSION

This project effectively illustrates the viability of utilising embedded machine learning to identify early symptoms of heart attacks by analysing real-time physiological parameters. A test accuracy of 96.64% was attained by a lightweight neural network model that was trained on a balanced synthetic dataset, suggesting that it exhibited strong predictive performance. The model was converted to TensorFlow Lite format and optimized, which enables deployment on low-power embedded hardware, such as Arduino Nano 33 BLE Sense Rev2. The system offers dependable and efficient real-time classification appropriate

References

[1] Nashif, S., Raihan, M. R., Islam, M. R., & Imam, M. H. (2018). Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system. World Journal of Engineering and Technology, 6(4), 854–873. https://doi.org/10.4236/wjet.2018.64057

[2] Uma, S., & Eswari, R. (2021). Accident prevention and safety assistance using IoT and machine learning. Journal of Reliable Intelligent Environments, 7(2), 185–198. https://doi.org/10.1007/s40860-021-00136-3

[3] Bharti, R., Khamparia, A., Shabaz, M., Dhiman, G., Pande, S., & Singh, P. (2021). Prediction of heart disease using a combination of machine learning and deep learning. Computational Intelligence and Neuroscience, 2021, Article ID 8387680. https://doi.org/10.1155/2021/8387680

[4] Zhang, D., Chen, Y., Chen, Y., Ye, S., Cai, W., Jiang, J., Xu, Y., Zheng, G., & Chen, M. (2021). Heart disease prediction based on the embedded feature selection method and deep neural network. Journal of Healthcare Engineering, 2021, Article ID 6661866. https://doi.org/10.1155/2021/6661866

[5] Elwahsh, H., El-Shafeiy, E., Alanazi, S., & Tawfeek, M. A. (2021). A new smart healthcare framework for real-time heart disease detection based on deep and machine learning. Journal of Healthcare Engineering, 2021, Article ID 9983654. https://doi.org/10.1155/2021/9983654

[6] Alnashwan, R., Saad, A., Qudaih, H., & Albraheem, L. (2019). IoT-based accident prevention system using machine learning techniques. International Journal of Computer Applications, 182(47), 1–5. https://doi.org/10.5120/ijca2019918220

[7] Zhu, L., Xiao, Y., & Li, X. (2021, January 15–17). Hybrid driver monitoring system based on Internet of Things and machine learning. In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 484–488). IEEE. https://doi.org/10.1109/ICCECE51280.2021.9342092

[8] Sharma, P., & Sood, N. (2020). Application of IoT and machine learning for real-time driver monitoring and assisting device. International Journal of Engineering and Advanced Technology, 9(3), 224–229. https://doi.org/10.35940/ijeat.C5047.029320

[9] Babu, C. V. S., A. N. S., & M. V. P. (2021). IoT-based smart accident detection and alert system. In Handbook of Research on Deep Learning Techniques for Cloud-Based Industrial IoT (pp. 1–20). IGI Global. https://doi.org/10.4018/978-1-7998-8591-2.ch001

[10] Zhu, L., Xiao, Y., & Li, X. (2020). Hybrid driver monitoring system based on Internet of Things and machine learning. IEEE Access, 8, 107562–107582. https://doi.org/10.1109/ACCESS.2020.3001149

[11] Arduino. (n.d.). Arduino® Nano 33 BLE Sense Rev2 - Description target areas. Arduino Docs. Retrieved September 19, 2025, from https://docs.arduino.cc/resources/datasheets/ABX00069-datasheet.pdf

[12] Arduino. (n.d.). Arduino® Nano 33 BLE Sense Rev2 - Full pinout. Arduino Docs. Retrieved September 19, 2025, from https://docs.arduino.cc/resources/pinouts/ABX00069-full-pinout.pdf

[13] Last Minute Engineers. (n.d.). MAX30102 pulse oximeter & heart rate sensor Arduino tutorial. Last Minute Engineers. Retrieved September 19, 2025, from https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/

[14] Seeed Studio. (n.d.). Grove - GSR sensor. Seeed Studio. Retrieved September 19, 2025, from https://wiki.seeedstudio.com/Grove-GSR_Sensor/

[15] Robu.in. (n.d.). SmartElex High Precision Temperature Sensor TMP117. Robu.in. Retrieved September 19, 2025, from https://robu.in/product/smartelex-high-precision-temperature-sensor-tmp117