


Article

Machine Learning Models for Emotion Recognition in Embedded Systems Based on Physiological Data

Šarūnas Kilius^{1,*}, Ričardas Gudonavičius¹, Darius Gailius¹, Mindaugas Knyva¹, Pranas Kuzas¹ ,
Darius Andriukaitis¹ , Gintautas Balčiūnas², Asta Meškuotienė²  and Justina Dobilienė² 

¹ Department of Electronics Engineering, Kaunas University of Technology, Studentu Str. 50-457, 51368 Kaunas, Lithuania; ricardas.gudonavicius@gmail.com (R.G.); darius.gailius@ktu.lt (D.G.); mindaugas.knyva@ktu.lt (M.K.); pranas.kuzas@ktu.lt (P.K.); darius.andriukaitis@ktu.lt (D.A.)

² Metrology Institute, Kaunas University of Technology, Studentu Str. 50-454, 51368 Kaunas, Lithuania; gintautas.balciunas@ktu.lt (G.B.); asta.meskuotiene@ktu.lt (A.M.); justina.dobilienė@ktu.lt (J.D.)

* Correspondence: sarunas.kilius@ktu.lt

Abstract

The increasing prevalence of work-related stress requires advanced, non-intrusive physiological monitoring solutions. As conventional methods are often impractical for continuous, real-world applications, this study investigates the deployment of artificial intelligence models on embedded systems for real-time emotion recognition from physiological signals. The study identified critical constraints for embedded implementation, including model size and memory capacity. An evaluation of various machine learning algorithms revealed that, while models like K-Nearest Neighbors (KNN) achieve high accuracy (88.8%), their excessive memory footprints make them unsuitable for resource-constrained hardware. Consequently, Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and recurrent neural network (RNN) architectures were deployed on an STM32F411 microcontroller, for which model compression proved essential. An experimental study validated the approach, achieving high recognition rates for pronounced emotions such as hatred (91%) and anger (85%), though with a lower accuracy for more subtle states. These results confirm the potential of embedded AI systems for physiological monitoring, highlighting the critical importance of feature selection and model compression for practical implementation.

Keywords: emotion recognition; embedded systems; physiological signals; machine learning

1. Introduction

The need for advanced physiological monitoring solutions has become increasingly apparent with the rising prevalence of chronic conditions such as cardiovascular diseases, diabetes, and stress-related disorders. The World Health Organization (WHO) [1] states that work-related stress is a major problem affecting the labor market worldwide. Work-related stress is the body's response to workplace demands and pressures that exceed an individual's capacity to cope. Stress, along with depression and anxiety, is the second most common workplace health problem in Europe, leading to increasing sick leave. A survey by the European Agency for Safety and Health at Work revealed that 51% of Europe's workers find stress "commonplace" in their workplace [2]. Furthermore, 66% of European employees report experiencing unhealthy levels of work-related stress [3]. The European Trade Union Institute (ETUI) estimates that work-related stress costs the EU over €100 billion annually [4,5]. This includes healthcare costs, lost productivity, and absenteeism



Academic Editor: Jichai Jeong

Received: 13 March 2026

Revised: 7 April 2026

Accepted: 9 April 2026

Published: 13 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

due to stress-induced illnesses [4]. Conventional monitoring approaches often fail to provide continuous assessment, timely interventions, and personalized insights. Traditional physiological monitoring systems typically rely on specialized medical equipment, limiting their accessibility and continuous application. AI-driven embedded systems address these limitations by enabling the intelligent processing of physiological signals, adaptive learning from user data, and context-aware health assessments.

Recent advancements in miniaturized sensors, low-power microprocessors, and efficient AI algorithms have accelerated the development of embedded physiological monitoring systems. Wearable sensors can monitor parameters such as blood pressure and support the management of conditions including epilepsy, diabetes, and cardiac and gastrointestinal disorders. Vital signs can now be monitored using sensors embedded in infusion pumps, chest bands, finger pulse oximeters [6,7], wrist-worn accelerometers (to measure movement in epilepsy patients) [8] and EEG sensors to measure brain activity. These technologies converge to create platforms capable of capturing, processing, and interpreting physiological signals with high fidelity while operating within the stringent constraints of embedded environments. The Internet of Medical Things (IoMT) paradigm has further enhanced these capabilities by establishing interconnected ecosystems of physiological monitoring devices that communicate with cloud services, healthcare providers, and other stakeholders [9].

Despite significant progress, the implementation of AI in embedded systems for physiological monitoring faces numerous challenges. These include hardware limitations, power consumption constraints, signal integrity issues, algorithmic complexity, and concerns related to privacy and security. Additionally, the accuracy and reliability of AI models in diverse real-world scenarios remain critical considerations for clinical adoption and user acceptance.

This paper presents a comprehensive methodology for the practical implementation of real-time emotion recognition on resource-constrained microcontrollers using physiological signals. Our main contribution is a structured workflow that systematically addresses the critical trade-off between model performance and the tight memory constraints of embedded hardware. The proposed solution uniquely integrates a multi-step process that includes: (1) the benchmarking of various machine learning architectures to identify optimal candidates; (2) the application of model compression as a crucial step to ensure deployability on the target microcontroller platform; and (3) empirical validation of the performance of compressed models under real-world conditions. This work demonstrates a promising and structured way to integrate sophisticated AI functions into everyday devices to continuously and unobtrusively monitor physiological signals.

The paper is structured as follows: Section 2 reviews the state of the art in physiological monitoring and the challenges of deploying models on embedded systems. Section 3 details the methodology, covering the dataset, data preprocessing, feature engineering, model training, and implementation on the microcontroller. Section 4 discusses the user-study results. Finally, Section 5 concludes the paper with a summary of the key findings.

2. Related Works

2.1. Challenges in Non-Intrusive Emotion Recognition via Computer Interfaces

One of the most critical factors influencing the performance of AI models in physiological emotion recognition is the quality and diversity of training data. Robust AI systems require extensive, high-resolution datasets that encompass a wide range of physiological responses across different individuals, contexts, and emotional states. Insufficient or biased data can lead to inaccurate predictions, reduced generalizability, and algorithmic bias. Since many people spend a lot of time working and relaxing at a computer, one simple way to monitor physiological signals is by using the computer mouse.

There are two main ways to monitor a person's emotional state using a computer mouse. The first is by analyzing mouse movements—for example, how fast or smoothly the person moves the cursor, how often they pause, or how accurately they click. Changes in these patterns can indicate stress, tiredness, or nervousness. The second method involves using built-in sensors to collect biometric data, such as skin temperature or pressure applied while holding the mouse. This allows for more direct insights into the user's physical condition. Both approaches can help AI systems detect emotional and physiological changes in a simple and non-intrusive way.

Lucia Pepa et al. [10] investigated stress detection through computer mouse and keyboard dynamics (K&MD), emphasizing the need for cost-effective, non-intrusive, and subject-independent systems suitable for real-world use. Their study addressed the scarcity of field research on K&MD-based stress recognition by conducting an in-the-wild experiment with 62 participants. Using a custom web application, participants performed eight computer tasks—including typing, the Tower of Hanoi, and memory games—under varying stress conditions such as time pressure and noise. Data included self-reported stress levels and detailed keystroke and mouse metrics. The authors applied Multiple Instance Learning with a Random Forest classifier using a 5 s sliding window and subject-independent cross-validation. The system classified stress into low, medium, and high levels with accuracies of 76% (keyboard) and 63% (mouse). Key discriminative features were dwell time, latency, and velocity, confirming the feasibility of K&MD for unobtrusive stress monitoring.

Mary Jane C. Samonte et al. [11] developed an affect detection model for gamers using mouse movement patterns during gameplay in *Torchlight 2*. The study aimed to identify mouse features linked to emotional states, evaluate different classifiers, and assess how sample size influences model performance. Data were collected from over 30 gamers, including mouse, facial, and bodily movement recordings. Mouse data—captured via MouseKeyLogger—were processed into 33 features such as click count and movement distance. Emotional labels were assigned through consensus by three psychology-trained annotators for 15 s segments. Four feature-based models were tested using Decision Tree, Random Forest, and J48 classifiers, with accuracy and kappa used for evaluation. The J48 classifier with Model 3 performed best, achieving an 88.23% accuracy and a kappa of 0.747. The most frequent affective state was “flow,” while mismatched game difficulty increased negative emotions, aligning with prior research on challenge–emotion balance.

T. Androutsou et al. present [12] a prototype of a smart computer mouse designed to monitor stress-related physiological signals during regular office work, without requiring users to wear extra devices or change their behavior. Their goal was to show that it is possible to collect reliable health data using a familiar everyday object—the computer mouse. The system integrates a photoplethysmography (PPG) sensor positioned on the side of the mouse to measure changes in blood volume from the user's thumb, and two galvanic skin response (GSR) electrodes placed on the top surface to detect variations in skin conductivity from the palm (Figure 1). These biological signals are processed locally by an embedded microcontroller, a 120 MHz ARM Cortex M3 on a Particle Photon board, housed inside the mouse, and the data were later transmitted wirelessly via an onboard Wi-Fi module. While the study did not test stress levels under controlled lab conditions, it successfully proved that the hardware works as intended. During the study, it was confirmed that the PPG signal can effectively capture changes in blood volume circulation, which are directly influenced by heartbeats. As a result, analyzing the PPG signal allows for the extraction of several key physiological measurements, including heart rate (HR) and heart rate variability (HRV). These indicators are strongly correlated with stress levels

and are widely recognized as reliable markers for assessing autonomic nervous system activity [13].



Figure 1. The developed smart mouse with PPG and GSR sensors [12].

Another study states that two key data points are used to analyze the PPG signal: the maximum point, or peak, and the minimum point, or valley [14]. A critical step in PPG processing is the detection of these peaks and valleys because it supports the estimation of the heart rate, heart rate variability, mental stress, and blood pressure to diagnose diseases. In this research, a PPG sensor with a Bluetooth module was integrated into a computer mouse, and 10 min recordings were collected from three participants: in the first dataset, the participant held the mouse and moved it slowly while performing a few simple tasks; in the second dataset, the participant used the mouse to read an electronic document, resulting in faster mouse movements; in the third dataset, the participant browsed the internet, moving the mouse quickly and clicking frequently. To process the data, the researchers used Adaptive Threshold (ADT) and Local Maximum and Minimum detection (LCM) algorithms, along with a newly developed method that detects peaks based on signal phases. The first phase is used to identify pairs of signal peaks and valleys using an adaptive peak detection algorithm. The second phase identifies and filters random errors. The results show that the recorded signals contained motion artifacts and high-amplitude baseline drift. However, the new algorithm successfully identified both early and delayed peaks and removed false detections. It corrected 103 peak values in the third dataset, which had high noise levels, while the ADT and LCM algorithms corrected only 22 peaks. These findings demonstrate that, with an appropriate algorithm, it is possible to overcome significant challenges in PPG peak detection under complex conditions.

2.2. Challenges for Deploying Neural Networks on Microcontrollers

One of the main challenges when deploying a neural network into microcontroller code is the limited hardware resources. Several studies have addressed this issue. For example, N. J. Cotton et al. present a practical approach for deploying neural networks on microcontrollers, focusing on implementation details such as activation functions optimized for faster computation [15]. Another study by P. E. Novac examines the trade-offs between neural network accuracy and the microcontroller's power consumption and memory usage [16]. These works highlight that the key challenges in implementing neural networks on microcontrollers are the speed of neuron computation, the memory footprint, the power consumption, and the overall accuracy of the model.

J. K. Ranbirsingh et al., in their research [17], focus on distributed deep learning models for Human Activity Recognition (HAR). The researchers implemented a distributed Long Short-Term Memory (LSTM) model—a recurrent neural network well suited for sequential data—using the TensorFlow framework and Python 3, both of which support distributed

architectures. The study combines concepts from deep learning and distributed systems, conducting experiments on two distinct hardware setups: a 16-node Raspberry Pi cluster and a multi-core Intel Xeon CPU system. The UCI HAR dataset was used for training, and model performance was assessed based on execution time and prediction accuracy across different configurations of layers and hidden units. The initial results indicated that a 3-layer distributed LSTM model offered the best trade-off between accuracy and training time. This configuration was further analyzed using two training strategies: a parameter server method on the Raspberry Pi cluster and an all-reduce synchronous stochastic gradient descent method on the Xeon system. Notably, the Raspberry Pi cluster slightly outperformed the Xeon system in prediction accuracy—by approximately 1% across varying node counts (32, 64, 128)—but required nearly twice the training time. However, both systems experienced a significant drop in accuracy (to ~72%) when the node count increased to 256.

N. Attaran states that personalized wearable biomedical systems enable the acquisition of various physiological and behavioral data that can be used to make general inferences about the human state [18]. These systems need to process parallel streams of multi-physiological data in real time and within a limited power budget. The core objective of the research was to develop an efficient local processor capable of processing multi-modal physiological signals using feature extraction and machine learning classifiers. During the experiment, wearable sensors in a LifeShirt captured physiological data from 15 participants. Based on the optimal feature set, specialized hardware processors for the Support Vector Machine (SVM) and KNN classifiers were designed and implemented. These processors were realized on ASIC (Application-Specific Integrated Circuit) and FPGA (Field-Programmable Gate Array) platforms, and their performance was compared against general-purpose embedded platforms such as Raspberry Pi 3B and NVIDIA Jetson TX1/TX2 GPUs.

Table 1 presents a comparison of the Energy–Delay Product (EDP) across all evaluated platforms. The ASIC implementation demonstrates a substantially lower EDP than alternative platforms, a crucial advantage for biomedical applications where both rapid decision-making and minimal energy consumption are important. Specifically, the ASIC achieves 16× and 100× lower EDP values than FPGA for KNN and SVM implementations, respectively. Additionally, the ASIC shows superior energy efficiency, outperforming FPGA by factors of 11× and 42× for KNN and SVM classifiers. While ASIC implementations offer remarkable energy efficiency improvements, its practical deployment may be constrained by higher development costs and longer time-to-market considerations. In contrast, the FPGA solution maintains the second-lowest EDP for stress detection and offers advantages such as re-programmability and lower development costs when compared to ASIC implementation.

Table 1. Hardware results from running stress detection applications on different processing platforms [18].

Processor	Clock, MHz	Power, mW	Throughput, dec/s	Energy, mJ	Energy Efficiency, dec/s/watt	Energy Efficiency Improvement (Over Baseline)
ARM A53	900	1480	2	746.36	1.33	1×
TX2 GPU	845	2120	130.54	16.23	61.58	46×
TX1 GPU	998	2430	225	10.76	92.89	69×
Artix-7 100T FPGA	200	728	195,121	0.0035	268,024	200,044×

Table 1. Cont.

Processor	Clock, MHz	Power, mW	Throughput, dec/s	Energy, mJ	Energy Efficiency, dec/s/watt	Energy Efficiency Improvement (Over Baseline)
ASIC	250	76.69	243,902	0.0003	3,180,368	2,373,712×

M. Trabelsi Ajili et al., in their research [19], present an FPGA-based acceleration for DeepSense, employing a hardware/software (HW/SW) co-design approach using the Xilinx Vitis AI framework and its Deep Learning Processing Unit (DPU). A new methodology was introduced to adapt DeepSense and its components to overcome the architecture's drawbacks and limitations within the Vitis AI framework, especially for time-series multimodal neural networks. CNN layers were accelerated on the DPU, while unsupported RNN (Gated Recurrent Unit—GRU) layers and the final fully connected (FC) layer were implemented on the CPU using TensorFlow Lite. This partitioning was chosen due to DPU limitations regarding RNNs and the significant data transfer overhead associated with running small FC layers on the DPU. The implementation was quantitatively evaluated on Xilinx Zynq UltraScale+ MPSoC boards, demonstrating significant performance gains over software-only baselines. The hybrid CPU-FPGA system achieved up to a 2.5× reduction in inference latency and a 5.2× improvement in energy efficiency. Despite the 8-bit quantization, the model maintained an acceptable accuracy of 0.95.

A comparative analysis of the current state of the art is presented in Table 2 with focus on the non-intrusive emotion and stress recognition methods. The modality, algorithm and target hardware platform definition along with the reported accuracy are presented. The identified limitations and gaps regarding the existing methods and the one proposed by the authors have been identified and addressed.

Table 2. Comparison of the proposed embedded emotion recognition system with related works.

Reference	Modality/Data Source	ML Algorithm	Target Hardware Platform	Reported Accuracy	Identified Limitation/Gap
Pepa et al. [10]	Mouse & Keyboard dynamics	Random Forest	Standard PC	63% (mouse)–76% (keyboard)	Requires continuous physical interaction; moderate accuracy.
Samonte et al. [11]	Mouse dynamics (gaming)	J48 Decision Tree	Standard PC	88.23%	High accuracy but relies on specific high-intensity tasks (gaming); not embedded.
Attaran et al. [18]	Multi-physiological (wearable)	KNN, SVM	ASIC, FPGA, Jetson TX1/TX2	~95.8%	Excellent performance but relies on highly specialized, expensive, and power-hungry hardware.
Trabelsi Ajili et al. [19]	Multi-modal signals	CNN + RNN	CPU-FPGA architecture	~95.0%	Deep learning is utilized, but hardware requires significant physical space and cost.
Proposed Work	PPG (smart mouse sensor)	LSTM (Optimized & Compressed)	STM32F411 microcontroller	74.8–91.0% (emotion-dependent)	Bridges the gap by successfully deploying deep learning on highly resource-constrained, low-cost Edge-AI hardware.

As demonstrated in Table 2, while the existing studies have achieved commendable accuracies (e.g., 88% to 95%) using various physiological and behavioral modalities, they predominantly rely on standard desktop computers or high-performance hardware accelerators like FPGAs and ASICs. These platforms, while powerful, are often impractical

for ubiquitous, low-cost integration into everyday office peripherals. The critical research gap lies in translating these complex deep learning architectures into highly constrained Edge-AI environments. To the best of the authors' knowledge, our proposed methodology addresses this exact limitation. By aggressively compressing a recurrent neural network (LSTM) and deploying it onto an STM32F411 microcontroller (limited to 512 KB Flash), we demonstrate in the proposed method that it is feasible to maintain competitive emotion recognition accuracy (up to 91% for specific pronounced emotions) while strictly operating within the memory and power constraints of a low-cost embedded system.

3. Materials and Methods

3.1. Dataset Description

The dataset used in this study originates from a prior experimental investigation that aimed to determine whether physiological signals can reflect consistent patterns corresponding to different emotional states experienced by an individual. Specifically, the research focused on identifying recurring physiological signal patterns associated with a set of eight states: seven specific emotions (anger, hatred, sadness, platonic love, romantic love, joy, respect) and one neutral/baseline state (no emotion).

The dataset generated and analyzed during the current study is not publicly available but is available from the corresponding author on reasonable request. All methods were carried out in accordance with relevant guidelines and regulations. Informed consent was obtained from all subjects involved in the study. Prior to their participation, all individuals were provided with a detailed explanation of the study's objectives, procedures, and the nature of the data being collected. All participants provided verbal consent to participate.

The dataset comprises 160 individual recordings, capturing more than 5,760,000 heartbeats in total. These data were collected over a 20-day period. During each experimental session, 30 min of physiological data were continuously recorded, allowing for the detailed temporal analysis of heart-related signals across varying emotional conditions.

The data were collected from 12 healthy individuals (see Table 3). The sample was composed of five female participants, aged between 25 and 35 years, with a mean age of 29.6 years, and seven male participants, whose mean age was 30 years, ranging from 24 to 36 years. Each subject participated in a single experimental session, during which they remained seated while watching and listening to video material designed to elicit specific emotional responses.

Table 3. Participant characteristics.

Participant Group	Number of Individuals	Age Range (Years)
Female	5	25–35
Male	7	24–36

Figure 2 presents an example of a photoplethysmography (PPG) signal from the first subject of the first dataset, illustrating the nature of raw physiological data and the typical artifacts encountered in real-world acquisition scenarios. In the top panel, a 500 s segment of the PPG signal is shown. This longer recording period demonstrates the presence of low-frequency baseline drift, which is typically caused by human movement or varying sensor-skin contact pressure. The bottom panel shows a zoomed-in 4 s interval of the same signal. In this magnified view, high-frequency noise components become clearly visible.

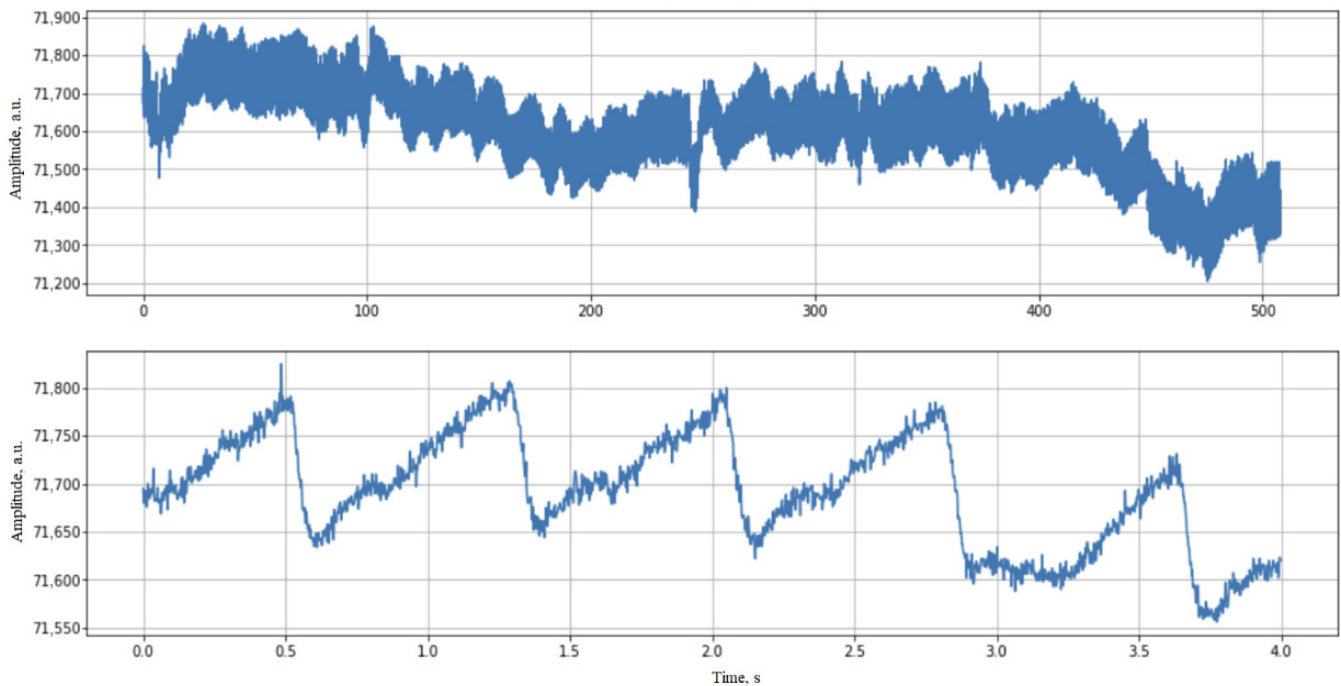


Figure 2. An example of a photoplethysmography (PPG) signal: (**top**) panel—500 s segment of the PPG signal; (**bottom**) panel—a zoomed-in 4 s signal.

3.2. Parameter Analysis

Physiological signal parameters can be classified into two main groups: biomedically significant parameters and signal quality parameters. Biomedically significant parameters are those directly related to human physiology and may be used to assess specific health conditions, emotional states, or other organism-related aspects. Signal quality parameters, on the other hand, are intended to evaluate the integrity, stability, and noise level of the physiological signal, all of which can significantly impact the accuracy and reliability of subsequent data analysis.

Key physiological parameters that can be derived from the PPG signal:

- RR interval: In PPG signals, this interval refers to the time elapsed between two consecutive systolic peaks (often termed the peak-to-peak interval), reflecting the heart's pumping cycle (as visually annotated in Figure 3).
- Heart rate (HR).
- Heart rate variability (HRV).
- Blood pressure: systolic (SBP) and diastolic (DBP).
- Respiratory rate.

Figure 3 illustrates two photoplethysmography (PPG) signal pulses, highlighting three key measurable parameters. These three primary parameters provide a basis for calculating the additional physiological parameters mentioned earlier.

To maintain the focus of this research work on machine learning optimization and embedded hardware deployment, the standard mathematical formulas for these physiological parameters (e.g., calculating mean blood pressure from systolic and diastolic peaks) are omitted. For this study, the parameter values were statistically aggregated (e.g., averaged) over the epochs corresponding to the duration of the specific emotional stimuli. While this approach captures the sustained physiological state associated with the target emotion, the authors acknowledge that the length of the signal window plays a critical role in emotion recognition. Short windows are highly sensitive to noise and motion artifacts, whereas excessively long windows may fail to capture transient emotional spikes. Optimizing the

temporal window size for feature extraction was outside of the scope of this hardware-focused study, but it remains a crucial direction for future work to further enhance the system's sensitivity.

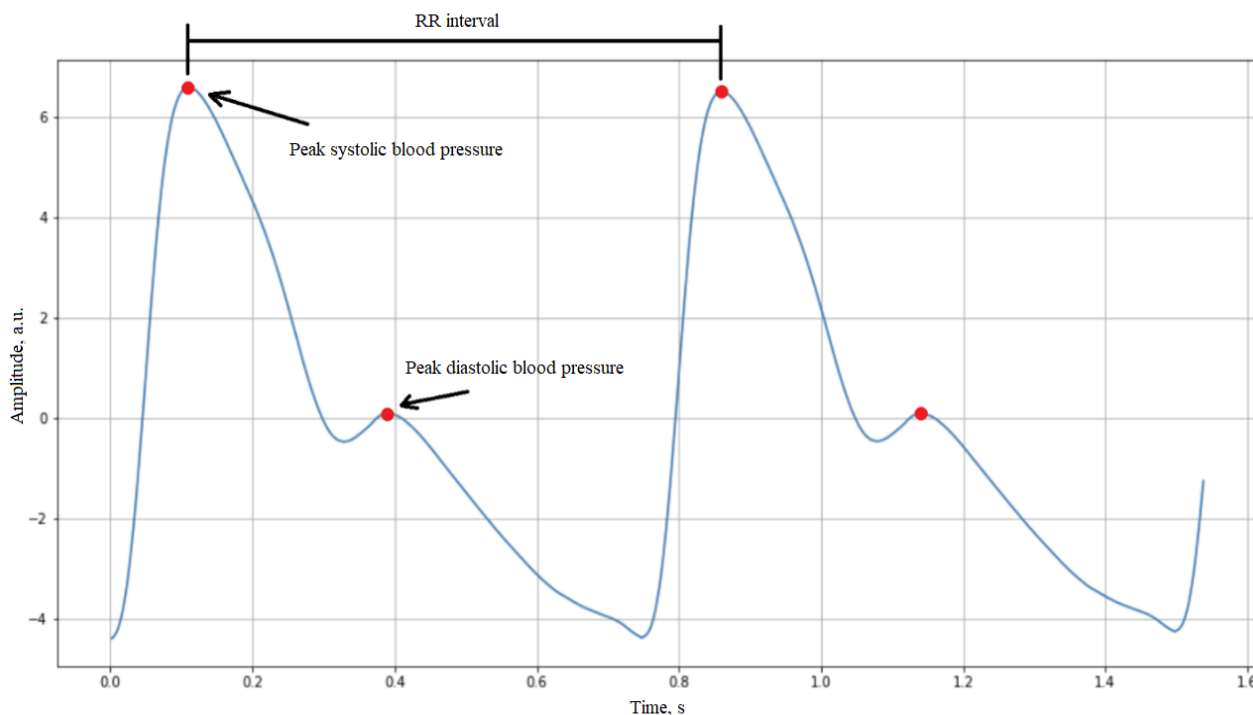


Figure 3. Example of PPG signal pulses with annotated parameters.

3.3. Data Preparation

Before model training, the essential preprocessing steps included organizing the datasets, removing signal noise, and eliminating corrupted components. This ensures that only high-quality, analyzable signals are used for further processing and parameter extraction.

Digital filtering techniques were used to reduce unwanted signal components and improve signal quality. In this work, a Butterworth Infinite Impulse Response (IIR) filter was selected due to its computational efficiency. Compared to Finite Impulse Response (FIR) filters, the IIR filter requires fewer coefficients to achieve an equivalent output response, which is particularly important in systems where computational resources are limited.

High-pass filtering was applied with a cutoff frequency of 0.5 Hz to remove baseline drift caused by low-frequency respiratory movements and blood flow artifacts. Although noise components generally lie within the 0 to 0.67 Hz frequency band, a 0.5 Hz cutoff was chosen to retain potentially useful signal components near 0.6 Hz, balancing noise suppression with signal preservation.

A 3rd-order low-pass filter with a cutoff frequency of 4 Hz was also implemented. Its main purpose was to suppress motion artifacts and eliminate the 50 Hz interference typically emitted by electrical devices. The attenuation characteristics of this filter were chosen to preserve the main physiological components of the signal, which are typically found between 0.6 Hz and 2 Hz. In addition, the filter parameters should ideally be adjusted to consider individual characteristics such as age, weight, height, and acute medical conditions, as these factors can significantly affect heart rate and signal morphology.

After filtering, the PPG signal was inverted to ensure that its peaks correspond to the systolic blood pressure maxima. This step facilitates the accurate detection of peak values using the peak detection algorithms. The first step of parameter extraction involved the

detection of systolic (SBP) and diastolic (DBP) blood pressure peaks. Based on these initial parameters, other required parameters were subsequently calculated.

Figure 4 shows the points where DBP and SBP peaks are detected within the PPG signal sourced from the dataset. It is evident that the signal components were filtered, as significant distortions, which were present in Figure 2, are no longer observable.

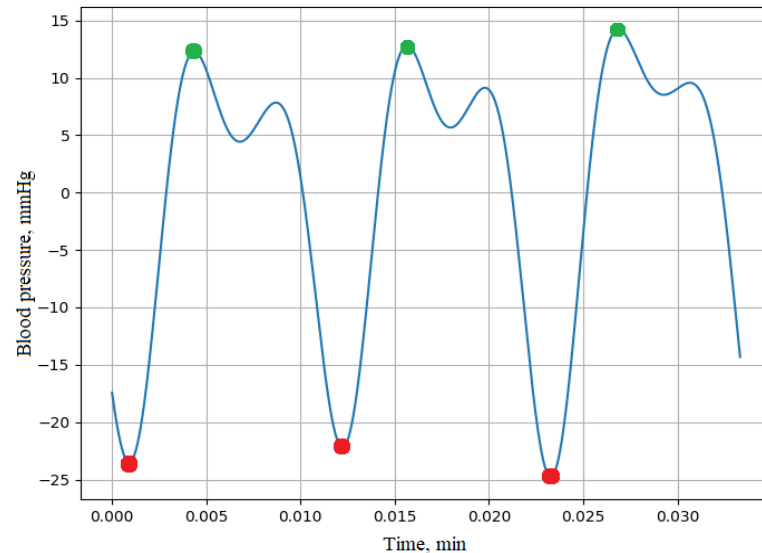


Figure 4. Peak blood pressure values (diastolic blood pressure—DBP in red, systolic blood pressure—SBP in green).

Following initial data preparation, normalization was performed. This is a data processing technique aimed at rescaling the data so that different features (attributes) are on a similar scale or contribute more equally. Normalization is particularly important for many machine learning algorithms, as some are sensitive to variations in feature scales; this process mitigates such influence.

For the normalization process, the StandardScaler from the Scikit-learn v1.2 library was used to apply the normalization. StandardScaler normalizes data by transforming them into Z-scores, following a standard normal distribution. This means that the mean of each feature will be 0, and its standard deviation will be 1. This transformation is performed according to the formula

$$z = \frac{(x - \mu)}{\sigma}, \quad (1)$$

where x is the original feature value; μ is the mean of the feature; and σ is the standard deviation of the feature.

Normalization reduces differences in feature scales, which can help improve model performance, reduce training time, and make results more interpretable. Furthermore, it is particularly crucial when an algorithm relies on feature distances (e.g., k-means clustering or the K-Nearest Neighbors algorithm), as these algorithms are highly sensitive to variations in feature scales.

In the next step, cross-validation was used to assess the generalization abilities of models when they are applied to new data. The dataset was partitioned into N distinct subsets, commonly referred to as folds. The value of N was predetermined based on the size and characteristics of the dataset. In each iteration of the cross-validation process, the model was trained on $N - 1$ folds and tested on the remaining fold. This procedure was repeated N times, ensuring that each fold was used exactly once as the test set.

For each iteration, the performance metrics—in particular classification accuracy, precision, recall, and $F1$ -score—were computed. After all iterations, the metric values

were averaged to produce a single performance estimate that reflects the model's ability to generalize across the entire dataset. This methodology can be mathematically described as

$$\bar{M} = \frac{1}{N} \sum_{i=1}^N M_i, \quad (2)$$

where \bar{M} is the average performance metric (e.g., accuracy); N is the number of folds; and M_i is the metric obtained in the i -th iteration.

Using cross-validation instead of a single train–test split ensures that all data points are used for both training and validation, which is particularly important in studies with limited sample sizes, as is often the case in physiological signal analysis. This helps minimize variance in performance estimation and supports more reliable conclusions regarding the model's effectiveness.

3.4. Parameter Importance Evaluation

In this study, the importance of different parameters was analyzed using Random Forest regressors for each individual emotion. During this process, a Random Forest model was trained to predict emotional states based on the provided physiological data. This method enabled the identification of the most relevant features for each specific emotion and provided insights into their utility in emotion recognition systems.

The examination of the most significant parameter for each emotion revealed that the physiological differences among emotional states are diverse and complex. This finding contributes to a deeper understanding of the physiological underpinnings of emotions and their recognition. Notably, the standard deviation and RR interval emerged as highly influential parameters for most emotions, indicating that these two physiological markers are particularly critical for accurately distinguishing and interpreting different emotional states (see Figure 5).

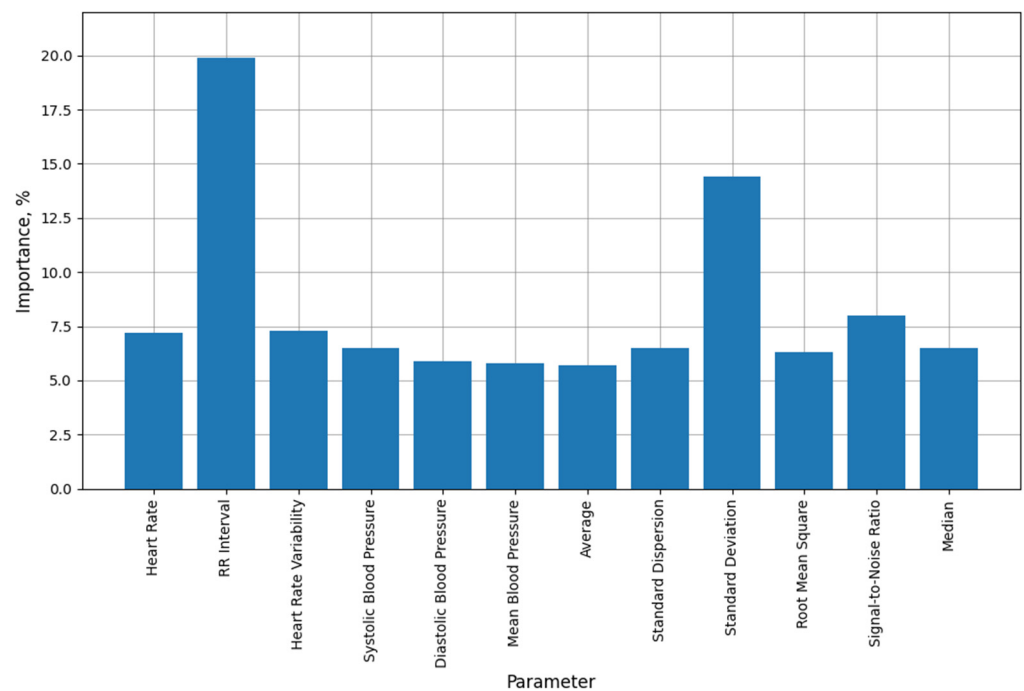


Figure 5. Example of parameter importance for emotion: joy.

In addition, heart rate variability (HRV) and heart rate (HR) themselves demonstrated a substantial relevance for certain emotions, making them important criteria for identifying

these emotional states. These metrics provide valuable temporal insights into autonomic nervous system dynamics, which are closely linked to emotional arousal and regulation.

On the other hand, the blood pressure-related parameters—such as systolic, diastolic, and mean arterial pressure—were found to have a lower overall importance in the emotion recognition process. However, for certain emotions, these parameters may still carry a degree of predictive value. This suggests that, although blood pressure is physiologically associated with emotional responses, its influence on recognition accuracy may be more limited compared to other features such as HRV or RR interval.

The analysis presented in Table 4 highlights the most significant physiological features and their corresponding importance percentages for each emotion. Notably, the standard deviation and RR interval consistently emerged as key indicators across multiple emotional states. For instance, the RR interval was the most important parameter for emotions such as anger (16.6%), platonic love (16.2%), romantic love (17.2%), joy (19.9%), and respect (21.5%), suggesting that heart rate dynamics play a crucial role in distinguishing these emotions. Similarly, the standard deviation was most relevant for the “no emotion” state (18.3%) and hatred (15.7%), indicating the importance of signal variability. Additionally, sadness was the most strongly associated with the signal-to-noise ratio (16.1%), underscoring the relevance of signal quality. These findings emphasize that, while some parameters are broadly informative, the emotion recognition performance can be optimized by tailoring the feature sets to each specific emotion.

Table 4. Parameter importance for emotions.

Parameters	Parameter Importance, %							
	No Emotion	Anger	Hatred	Sadness	Platonic Love	Romantic Love	Joy	Respect
Standard Deviation	18.3	15.8	15.7	18.5	16.3	13.3	14.4	16.1
RR Interval	10.9	10.4	16.6	7.9	16.2	17.2	19.9	21.5
Signal-to-Noise Ratio	8.6	10.9	11.2	16.1	9.5	9	8	12.1
Standard Dispersion	7.5	7	9.8	8.5	7.4	8.6	6.5	4.9
Heart Rate Variability	9.1	7.3	6.1	8.2	7.7	6.4	7.3	5.6
Heart Rate	8.6	6.9	6.5	7.8	7.3	6.2	7.2	5.2
Average	10.5	8.9	5.4	5.6	6.8	5.3	5.7	7.5
Systolic Blood Pressure	5.2	6.1	5.4	6.8	4.9	11.8	6.5	6.7
Median	5.7	8.6	7.1	5.5	6.7	6.3	6.5	6.1
Diastolic Blood Pressure	3.8	6.2	7.1	6.8	6.5	8	5.9	5.8
Root Mean Square	5.7	6	4.6	4	5.6	4.1	6.3	4.3
Mean Blood Pressure	6.1	5.9	4.4	4.3	5.2	3.7	5.8	4.3

The dominance of the RR interval and the standard deviation (SD) as the most discriminative features is strongly supported by physiological and signal processing principles. Physiologically, the SD of the RR intervals is a primary time-domain measure of heart rate variability (HRV). HRV is directly modulated by the autonomic nervous system (ANS). High-arousal emotional states (such as anger or joy) trigger sympathetic nervous system activity, which typically shortens the RR interval and decreases the overall standard deviation of the rhythm.

Furthermore, from a signal processing perspective, these parameters exhibit high robustness against the specific noise profiles of a mouse-embedded PPG sensor. Real-world mouse usage introduces significant motion artifacts and variable skin contact pressure, which severely distort the amplitude of the PPG signal. Because the RR interval and SD are timing-based features (relying solely on the temporal detection of systolic peaks rather than absolute signal amplitude), they remain relatively stable even when the signal’s baseline wanders or its amplitude fluctuates. This explains why amplitude-dependent features, such

as estimated systolic and diastolic blood pressure, ranked substantially lower in importance compared to robust temporal metrics.

3.5. Model Development

In this study, several machine learning algorithms—Random Forest, SVM, KNN, MLP, RNN, LSTM, and CNN—were applied for emotion recognition. For each algorithm, a separate model was developed and trained using both a maximal set of parameters and a reduced, optimal subset. The primary focus of model optimization was on adjusting the architecture and tuning the hyperparameters, including the learning rate and regularization coefficient. Hyperparameter optimization was carried out using grid search, random search, and Bayesian optimization methods, aiming to enhance the models' performance for each algorithm.

To ensure the robustness and reproducibility of the models, a rigorous training protocol was implemented. The dataset was evaluated using a k-fold cross-validation strategy to avoid overfitting and ensure robust generalization. For deep learning architectures (MLP, CNN, LSTM, and RNN), given the multi-class nature of the emotion recognition task (eight different states), categorical cross-entropy was strictly applied as the loss function. Hyperparameter optimization (using grid search, random search, and Bayesian methods) yielded specific configurations for each architecture. The models were built using the Adam and RMSprop optimizers with learning rates ranging from 0.001 to 0.01. Specifically, the MLP model was trained for 200 epochs with a learning rate of 0.001. The CNN model was trained for 100 epochs using a batch size of 32 and a learning rate of 0.01, and including L1/L2 regularization (0.01) to reduce overfitting. The recurrent models (LSTM and RNN) used a time step sequence length of 8 and included dropout layers (set to 0.5) as an additional structural regularization measure. These precise configurations were chosen to balance classification accuracy with the tight computational and memory constraints of the target STM32F411 microcontroller.

The post-training comparison of model accuracy revealed a varying sensitivity to parameter optimization (see Figure 6). The Random Forest (RF1) model slightly improved from 86.5% to 86.7%, indicating robustness and low overfitting. RF2 demonstrated a significant accuracy increase from 48.3% to 57.0%, showing a strong dependence on parameter tuning. The KNN model maintained a stable 88.8% accuracy, regardless of parameter count. The MLP saw a minor decrease from 71.6% to 70.9%, suggesting moderate stability. LSTM accuracy declined from 79.7% to 74.8% when optimized, suggesting that the more complex configuration performed better. CNN and RNN models improved notably—from 57.0% to 61.2% and from 50.0% to 58.4%, respectively—demonstrating a reliance on proper parameter configuration. The SVM model slightly decreased from 12.0% to 11.0% and remained the least accurate overall.

To gain deeper insights into the classification dynamics and to evaluate the trade-offs between theoretical accuracy and embedded feasibility, confusion matrices were generated for the two most significant optimized models (Figure 7). The left matrix illustrates the performance of the K-Nearest Neighbors (KNN) model, which achieved the highest overall accuracy (88.8%) during initial benchmarking. As can be seen in the matrix, the KNN model successfully isolates the emotional states with minimal cross-talk, demonstrating strong diagonal alignment and only minor confusion between adjacent classes. However, due to its massive memory footprint (exceeding 134 MB), this model represents a theoretical upper bound rather than a practical embedded solution.

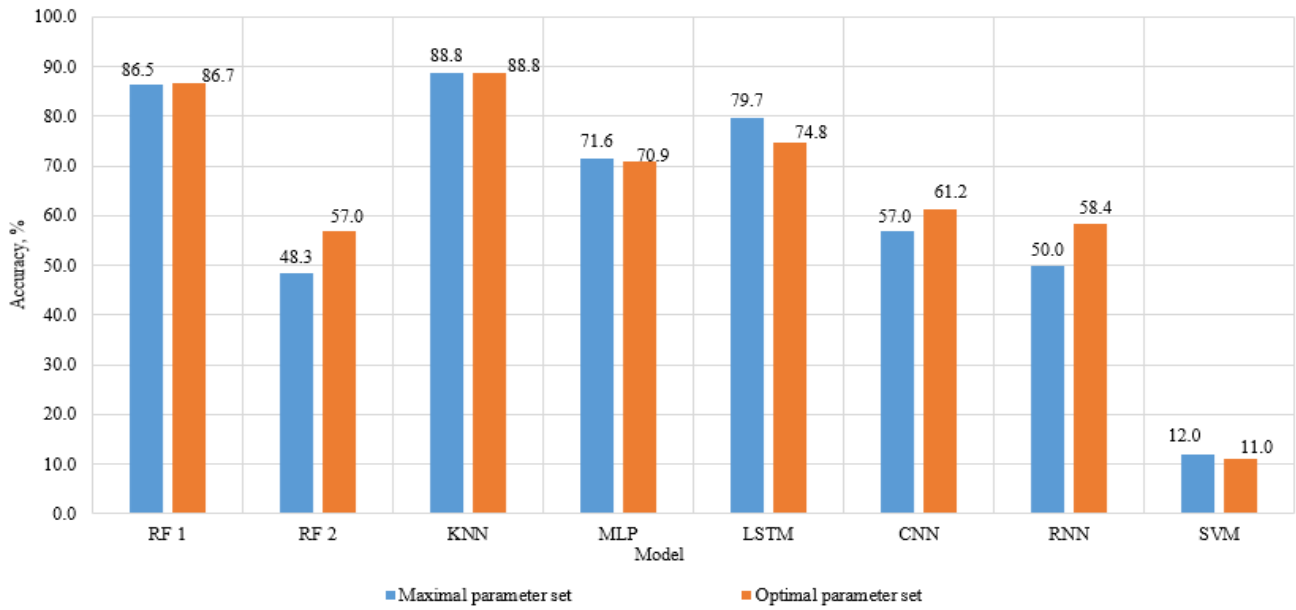


Figure 6. Comparison of model accuracy.

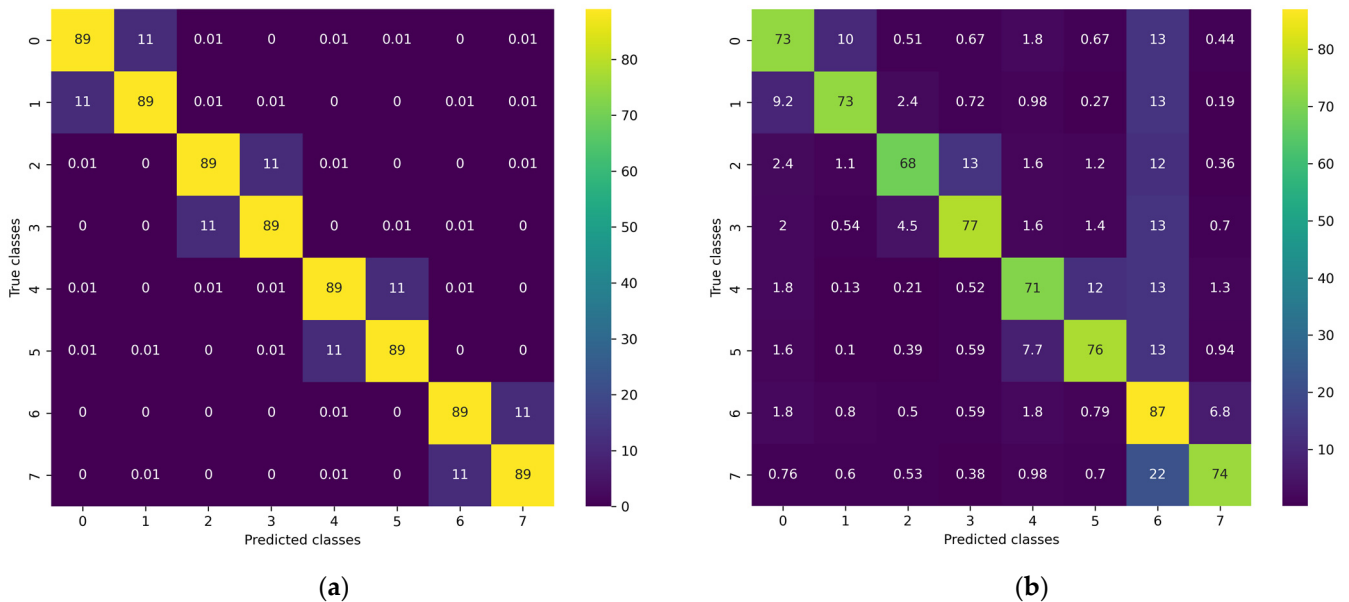


Figure 7. Confusion matrices for the optimized models: (a) KNN model demonstrating theoretical maximum accuracy; (b) LSTM model chosen for embedded deployment. Class labels correspond to: 0—no emotion, 1—anger, 2—hatred, 3—sadness, 4—platonic love, 5—romantic love, 6—joy, 7—respect.

Conversely, the right matrix illustrates the performance of the optimized LSTM model, which was selected for the final embedded deployment due to its superior compression capabilities. While the LSTM matrix still demonstrates solid diagonal alignment, it reveals the realistic challenges of physiological signal classification. The model successfully distinguishes pronounced high-arousal emotions but exhibits noticeable cross-talk between specific affective states (e.g., instances of Class 6 being confused with Class 7).

3.6. Embedded Implementation of Machine Learning Models

This section presents the deployment of trained models on an STM32F411 microcontroller. A photoplethysmography (PPG) heart rate sensor is attached to a computer mouse to measure the user’s pulse through the thumb. The analog signal is processed

by the microcontroller, which extracts features and performs emotion recognition. The resulting classification output is then transmitted to a computer via USB. The block diagram (Figure 8) illustrates the system architecture and signal flow used for model validation and real-time inference.

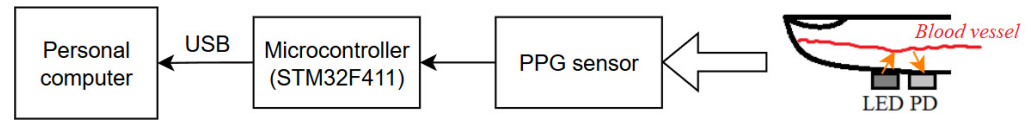


Figure 8. The block diagram of the system for PPG signal processing.

Due to the limited memory capacity of the selected STM32F411 microcontroller (Flash—512 KB, RAM—128 KB), the RF, KNN, and SVM models were discarded because their coefficient arrays exceeded the available memory, with sizes of 4 MB, 134.9 MB, and 34.5 MB, respectively.

Despite the exclusion of RF, KNN, and SVM models due to excessive coefficient array sizes, the remaining models (MLP, CNN, LSTM, and RNN) still exceeded the STM32F411 microcontroller's memory limits, requiring model compression. Figure 9 illustrates how memory usage varies depending on the applied compression level. A notable decrease in memory demand occurs when shifting from lossless to low compression. With low compression, memory usage was reduced to 196.58 KB for MLP, 258.54 KB for CNN, 639.84 KB for LSTM, and 1009.83 KB for RNN. However, LSTM and RNN still exceeded the 512 KB flash memory limit by 1.25× and 1.97×, respectively, and therefore required medium-level compression. Under high compression, the models consumed 100.78 KB (MLP), 236.67 KB (CNN), 411.62 KB (LSTM), and 403.66 KB (RNN), corresponding to approximately 19.7%, 46.2%, 80.4%, and 78.8% of the available flash memory. This demonstrates that compression is essential to enable model deployment on the microcontroller, and the required compression level depends on the specific model and the hardware constraints.

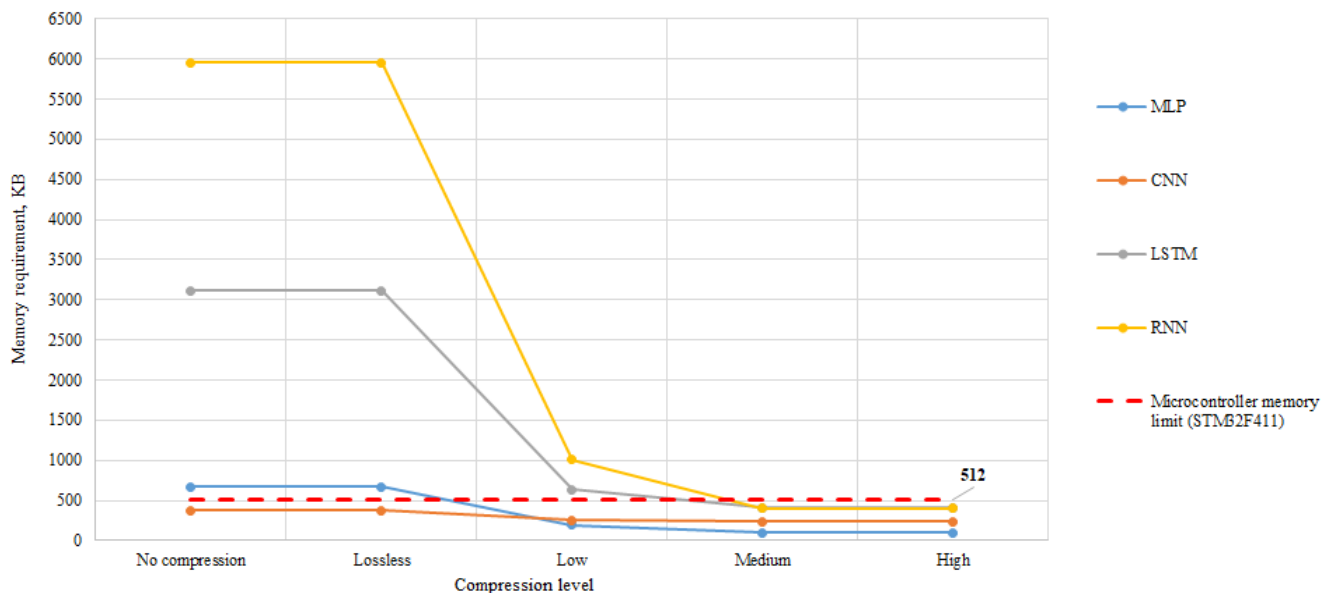


Figure 9. Comparison of memory requirements depending on compression level.

The LSTM model was made memory-feasible through targeted structural optimization. While the initial uncompressed LSTM model (3.1 MB) exceeded the hardware limits, its parametric nature allowed for aggressive quantization. As shown in Figure 9, by applying medium/high compression, the LSTM footprint was reduced to 411.62 KB. This footprint

occupies approximately 80.4% of the available microcontroller Flash memory, leaving sufficient space for the application logic and signal processing stacks. In contrast, the KNN model is inherently non-parametric and requires the storage of the entire training dataset to perform inference. In our case, this resulted in a model size exceeding 134.9 MB, which is approximately 263 times larger than the total Flash memory of the STM32F411 microcontroller (512 KB). Quantizing or compressing a KNN model is not feasible without drastically reducing the reference dataset, which would negate its accuracy advantage.

The model compression pipeline was implemented using the TensorFlow Lite (TFLite) converter framework. This process involves translating the high-level Keras models into an optimized flatbuffer format, which includes dead-code elimination, operator fusion, and graph optimizations specifically tailored for embedded ARM Cortex-M environments. To evaluate the impact of bit-precision on memory and performance, four distinct compression levels were defined:

- Lossless: Standard conversion without additional optimization parameters, maintaining original floating-point precision.
- Low: Post-training dynamic range quantization, which quantizes weights to 8-bit integers while keeping activations in floating-point during inference.
- Medium: Float16 quantization, reducing the precision of all weights and constants from 32-bit to 16-bit floating-point values.
- High: Full integer quantization, which maps all model tensors (including input and output) to 8-bit integer precision. This compression level utilized a representative dataset to calibrate the dynamic ranges of activations.

Figure 10 illustrates the changes in model accuracy as a function of the applied compression level. A slight decrease in accuracy is observed across all models as compression increases.

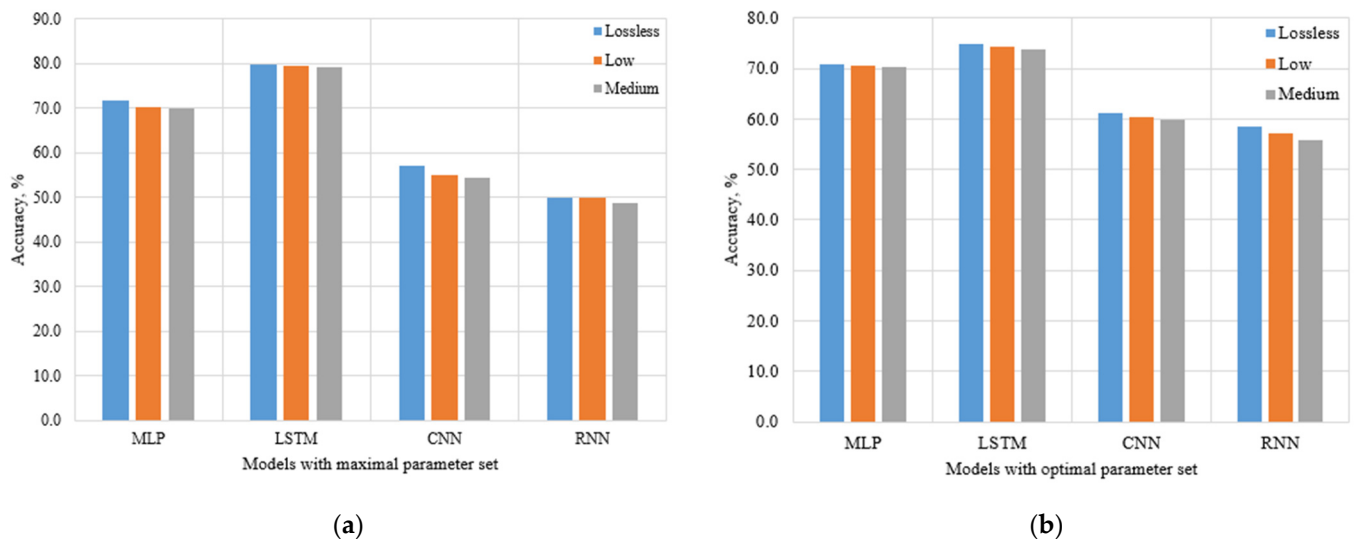


Figure 10. Comparison of model accuracy at different compression levels: (a) with maximum parameter set; (b) with optimal parameter set.

The observed accuracy variations across different compression levels provide a critical measure of the model's structural robustness. Specifically, the optimal LSTM model exhibited a marginal decrease in accuracy, dropping from 79.7% (uncompressed) to 79.0% (medium compression). This represents a mere 0.88% relative loss in predictive power, while simultaneously achieving a 7.57-fold reduction in the model size. Such a minor change falls within the expected variability for physiological signal processing, where the inherent noise floor of the PPG signal and the variance within the cross-validation folds

typically exceed this percentage. Furthermore, this trade-off is highly favorable for Edge-AI applications: the negligible loss in accuracy is a necessary and justified compromise to overcome the binary constraint of the STM32F411’s 512 KB Flash memory. Consequently, the compressed LSTM model remains largely comparable to its original version in practical use in real-world emotion recognition tasks, while enabling local, real-time inference on low-cost hardware.

4. Experiments and Results

The aim of the experiment was to recognize human emotions using the machine learning model that demonstrated the highest accuracy in prior testing. The experiment involved presenting video clips intended to induce the following emotional states: no emotion, anger, hatred, sadness, platonic love, romantic love, joy, and respect.

A total of four participants took part in the study—two women and two men (see Table 5). Each participant performed the same activity: while seated and holding a computer mouse embedded with a PPG (photoplethysmography) sensor, they watched video sequences specifically selected to evoke the target emotions. Each video lasted approximately five minutes, with a three-minute rest period between clips to minimize emotional overlap.

Table 5. Experiment participant characteristics.

Participant Group	Number	Age (Years)
Women	2	24
Men	2	24–25

The experiment assessed the model’s average emotion recognition performance based on physiological signals captured during the video sessions. The results presented in Figure 11 reflect the mean classification accuracy achieved across all participants and emotional categories.

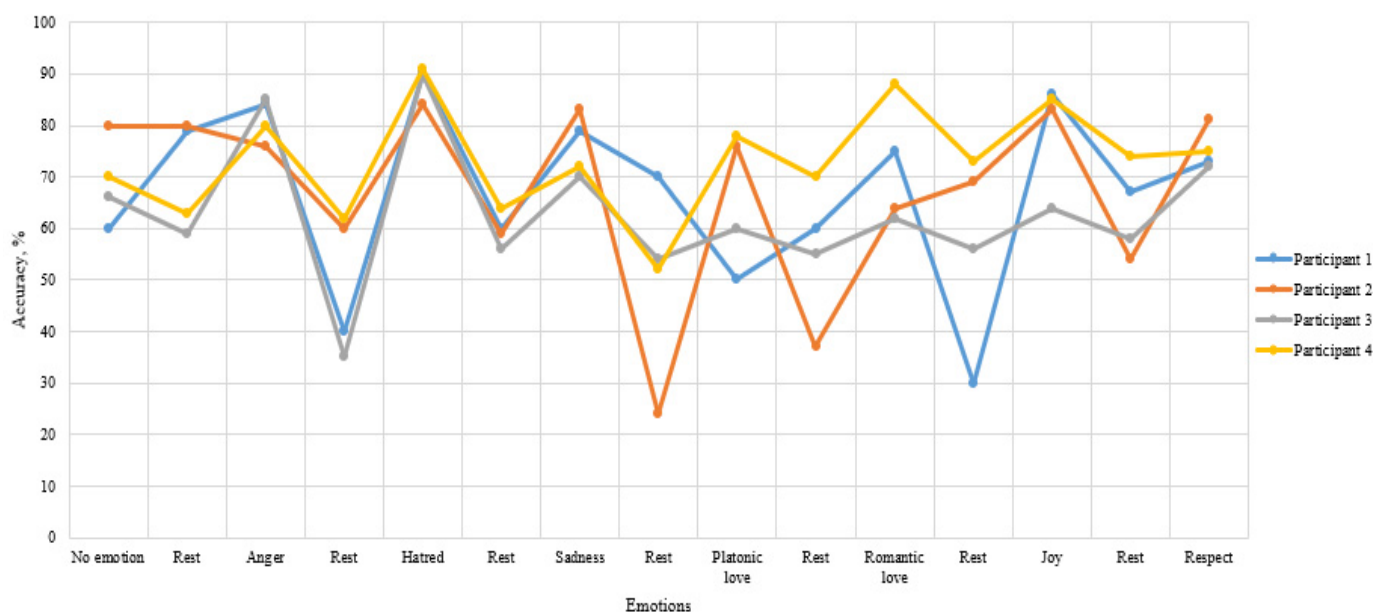


Figure 11. Experiment results.

The experiment evaluated the performance of an LSTM model in recognizing eight distinct emotions based on physiological responses from the participants. The analysis revealed that hatred was the most accurately recognized emotion across all participants,

with accuracy ranging from 84% to 91%. Anger also showed consistently high recognition rates (80–85%), making these two emotions the most reliably detected by the model. At the other end of the spectrum, platonic love, joy, and romantic love were among the least accurately recognized emotions. For example, Participant 1 showed only a 50% accuracy for platonic love, while Participant 3 achieved just 56% for joy and 58% for romantic love.

Comparing participants, Participant 4 demonstrated the most stable performance, with high accuracy in hatred (91%), anger (80%), and romantic love (88%). Participant 1 also performed well for hatred (90%) and anger (84%) but had significant drops for more subtle emotions like platonic love. Participant 2 showed similar trends, with high accuracy for hatred (84%) but notable difficulties with romantic love (65%).

The stark contrast in recognition rates between high-intensity emotions and subtle affective states is unlikely to stem from a severely imbalanced training dataset, as the stimulus epochs were proportionately distributed. Instead, this disparity underscores a fundamental physiological limitation of relying solely on PPG-derived features. From a psychophysiological perspective, emotions like hatred and anger are characterized by high arousal, triggering distinct and immediate sympathetic nervous system responses (“fight-or-flight”), which drastically alter the RR intervals and heart rate variability (HRV). Conversely, subtle states like platonic love or the resting baseline represent low-arousal conditions. The physiological responses during these states are often indistinguishable from one another using only peripheral cardiovascular metrics. Distinguishing such nuanced emotions inherently requires either more sensitive multi-modal sensing (e.g., EEG or galvanic skin response) or significantly longer observational windows to capture subtle autonomic shifts.

Overall, the LSTM model was effective in recognizing strong, physiologically pronounced emotions such as hatred and anger, but its performance declined with more nuanced affective states, particularly those related to affection or positive social bonding. This highlights areas for model refinement, especially in enhancing sensitivity to subtler emotional patterns.

5. Conclusions

The experiments highlight the importance of parameter optimization in machine learning models, with performance varying significantly across architectures. RF and CNN models achieved accuracy improvements (up to 8.7% for RF and 4.2% for CNN) when fine-tuned, while KNN maintained stable performance (88.8%) regardless of parameter adjustments. In contrast, SVM proved impractical due to excessive training time and poor accuracy (11–12%). These findings suggest that, while some models benefit from parameter tuning, others, like KNN, offer inherent stability, making them preferable for certain applications despite the computational trade-offs.

The initial model sizes significantly exceeded the typical memory constraints of microcontrollers—for instance, even the smallest Random Forest model required around 4 MB, while the KNN model exceeded 229 MB, far surpassing the common 512 KB limit. This highlighted the necessity of compression to enable deployment in embedded systems. Despite a noticeable reduction in accuracy for MLP, CNN, LSTM, and RNN models under higher compression levels, all models remained functionally effective. Notably, the LSTM model consistently achieved the highest accuracy across all compression stages, demonstrating its robustness and making it a strong candidate for real-time emotion recognition tasks on resource-constrained devices.

Emotion-specific analysis revealed disparities in recognition performance, with a high accuracy for distinct emotions like hatred (91%) and anger (85%) but lower detection rates for subtle states such as platonic love (50%). This variability suggests that expanding

training datasets and refining model architectures could improve the detection of complex affective states, including stress. While preliminary results validate the feasibility of physiology-based emotion recognition, further research is needed to enhance generalization across diverse emotional spectra. The study underscores the potential of AI in affective computing while highlighting the necessity of balancing computational efficiency, memory constraints, and predictive accuracy for real-world implementation.

While the proposed embedded ML models demonstrate promising accuracy, it is important to acknowledge the limitations of this study. The dataset was collected from 12 participants within a specific age group (24–36 years), which introduces a potential risk of bias toward this group. However, the risk of overfitting was actively mitigated through several strategies. First, the longitudinal nature of the data collection (over 20 days, yielding 5.7 million heartbeats) ensured high intra-subject variability. Second, rigorous feature selection reduced input dimensionality. Most importantly, the strict memory constraints of the STM32F411 microcontroller necessitated severe model compression. This compression inherently acts as a strong regularization technique, preventing the models from having the parameter capacity to overfit the training data. Therefore, this work serves primarily as a foundational proof-of-concept for embedded Edge-AI implementation in physiological monitoring. Future studies will require larger, more diverse cohorts to eliminate bias and further validate the generalizability of these highly compressed embedded models.

Author Contributions: Conceptualization, R.G. and Š.K.; methodology, R.G., M.K. and Š.K.; software, R.G. and Š.K.; validation, M.K. and D.G.; formal analysis, R.G., A.M. and J.D.; investigation, R.G. and Š.K.; writing—original draft preparation, R.G., Š.K. and P.K.; writing—review and editing, Š.K., D.A. and G.B.; visualization, R.G. and Š.K.; supervision, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: All authors do not have any financial or non-financial conflicts of interest.

References

1. Leka, S.; Griffiths, A.; Cox, T. *Work Organisation and Stress*; Protecting Workers' Health Series No. 3; Institute of Work, Health & Organizations: Nottingham, UK, 2003.
2. Swift, M. Stress Costs Europe over €600 Billion a Year. Available online: <https://meditationlifestyle.com/2022/06/11/stress-costs-europe-e94-billion-a-year/> (accessed on 10 June 2025).
3. Schipperen, T. Two-Thirds of European Employees Experience Excessive Work-Stress—Lepaya EN. Available online: <https://www.lepaya.com/blog/stress-at-work> (accessed on 10 June 2025).
4. Regional-Coverage—Health, Safety and Environment Review—European Report Quantifies Cost of Work-Related Stress | Health, Safety and Environment Review. Available online: <https://hsereview.com/regional-coverage/europe/european-report-quantifies-cost-of-work-related-stress> (accessed on 10 June 2025).
5. For the First Time, a European Report Puts a Price on Stress at Work | Etui. Available online: <https://www.etui.org/news/first-time-european-report-puts-price-stress-work> (accessed on 10 June 2025).
6. Matthew, P.; Mchale, S.; Deng, X.; Nakhla, G.; Trovati, M.; Nnamoko, N.; Pereira, E.; Zhang, H.; Raza, M. A Review of the State of the Art for the Internet of Medical Things. *Sci* **2025**, *7*, 36. [CrossRef]
7. Kazanskiy, N.L.; Butt, M.A.; Khonina, S.N. Recent Advances in Wearable Optical Sensor Automation Powered by Battery versus Skin-like Battery-Free Devices for Personal Healthcare—A Review. *Nanomaterials* **2022**, *12*, 334. [CrossRef] [PubMed]
8. Kusmakar, S.; Karmakar, C.K.; Yan, B.; O'Brien, T.J.; Muthuganapathy, R.; Palaniswami, M. Automated Detection of Convulsive Seizures Using a Wearable Accelerometer Device. *IEEE Trans. Biomed. Eng.* **2019**, *66*, 421–432. [CrossRef] [PubMed]
9. Ashfaq, Z.; Mumtaz, R.; Rafay, A.; Zaidi, S.M.H.; Saleem, H.; Mumtaz, S.; Shahid, A.; De Poorter, E.; Moerman, I. Embedded AI-Based Digi-Healthcare. *Appl. Sci.* **2022**, *12*, 519. [CrossRef]

10. Pepa, L.; Sabatelli, A.; Ciabattini, L.; Monteriu, A.; Lamberti, F.; Morra, L. Stress Detection in Computer Users From Keyboard and Mouse Dynamics. *IEEE Trans. Consum. Electron.* **2021**, *67*, 12–19. [[CrossRef](#)]
11. Samonte, M.J.C.; Veá, L.A.; San Jose, R.M.C.; John, V.; Lagoy, A.; Manlapid, C.E.P.; Martyn, P.; Perez, A. An Affect Detector Model For Gamers on a Role-Playing Game Through Mouse Movements. In *Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC)*; IEEE: Piscataway, NJ, USA, 2018; pp. 489–494.
12. Androutsou, T.; Angelopoulos, S.; Kouris, I.; Hristoforou, E.; Koutsouris, D. A Smart Computer Mouse with Biometric Sensors for Unobtrusive Office Work-Related Stress Monitoring. In *Proceedings of the 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*; IEEE: Piscataway, NJ, USA, 2021; pp. 7256–7259.
13. Betti, S.; Lova, R.M.; Rovini, E.; Acerbi, G.; Santarelli, L.; Cabiati, M.; Del Ry, S.; Cavallo, F. Evaluation of an Integrated System of Wearable Physiological Sensors for Stress Monitoring in Working Environments by Using Biological Markers. *IEEE Trans. Biomed. Eng.* **2018**, *65*, 1748–1758. [[CrossRef](#)] [[PubMed](#)]
14. Tran, T.V.; Chung, W.-Y. A Robust Algorithm for Real-Time Peak Detection of Photoplethysmograms Using a Personal Computer Mouse. *IEEE Sens. J.* **2015**, *15*, 4651–4659. [[CrossRef](#)]
15. Cotton, N.J.; Wilamowski, B.M.; Dundar, G. A Neural Network Implementation on an Inexpensive Eight Bit Microcontroller. In *Proceedings of the 2008 International Conference on Intelligent Engineering Systems*; IEEE: Piscataway, NJ, USA, 2008; pp. 109–114.
16. Novac, P.-E.; Castagnetti, A.; Russo, A.; Miramond, B.; Pegatoquet, A.; Verdier, F.; Castagnetti, A. Toward Unsupervised Human Activity Recognition on Microcontroller Units. In *Proceedings of the 2020 23rd Euromicro Conference on Digital System Design (DSD)*; IEEE: Piscataway, NJ, USA, 2020; pp. 542–550.
17. Ranbirsingh, J.K.; Kimm, H.; Kimm, H. Distributed Neural Networks Using TensorFlow over Multicore and Many-Core Systems. In *Proceedings of the 2019 IEEE 13th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip (MCSoc)*; IEEE: Piscataway, NJ, USA, 2019; pp. 101–107.
18. Attaran, N.; Puranik, A.; Brooks, J.; Mohsenin, T. Embedded Low-Power Processor for Personalized Stress Detection. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 2032–2036. [[CrossRef](#)]
19. Trabelsi Ajili, M.; Hara-Azumi, Y. Multimodal Neural Network Acceleration on a Hybrid CPU-FPGA Architecture: A Case Study. *IEEE Access* **2022**, *10*, 9603–9617. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.