Smart Health 28 (2023) 100390

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

An energy-efficient semi-supervised approach for on-device photoplethysmogram signal quality assessment

Mohammad Feli [a,*], Iman Azimi [b,a,c], Arman Anzanpour [a], Amir M. Rahmani [b,d,c], Pasi Liljeberg [a]

[a] Department of Computing, University of Turku, Turku, Finland
[b] Department of Computer Science, University of California Irvine, Irvine, USA
[c] Institute for Future Health, University of California, Irvine, USA
[d] School of Nursing, University of California, Irvine, CA, USA

ARTICLE INFO

Keywords:
Signal quality assessment
Photoplethysmography
Internet of Things
Real-time health monitoring
Semi-supervised learning

ABSTRACT

Photoplethysmography (PPG) is a non-invasive technique used in wearable devices to measure vital signs (e.g., heart rate). The method is, however, highly susceptible to motion artifacts, which are inevitable in remote health monitoring. Noise reduces signal quality, leading to inaccurate decision-making. In addition, unreliable data collection and transmission waste a massive amount of energy on battery-powered devices. Studies in the literature have proposed PPG signal quality assessment (SQA) enabled by rule-based and machine learning (ML)-based methods. However, rule-based techniques were designed according to certain specifications, resulting in lower accuracy with unseen noise and artifacts. ML methods have mainly been developed to ensure high accuracy without considering execution time and device's energy consumption. In this paper, we propose a lightweight and energy-efficient PPG SQA method enabled by a semi-supervised learning strategy for edge devices. We first extract a wide range of features from PPG and then select the best features in terms of accuracy and latency. Second, we train a one-class support vector machine model to classify PPG signals into "Reliable" and "Unreliable" classes. We evaluate the proposed method in terms of accuracy, execution time, and energy consumption on two embedded devices, in comparison to five state-of-the-art PPG SQA methods. The methods are assessed using a PPG dataset collected via smartwatches from 46 individuals in free-living conditions. The proposed method outperforms the other methods by achieving an accuracy of 0.97 and a false positive rate of 0.01. It also provides the lowest latency and energy consumption compared to other ML-based methods.

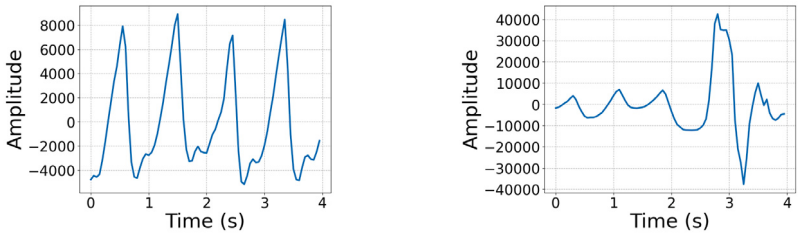
1. Introduction

Thanks to the advances in intelligent wearables and Internet-of-Things (IoT) technologies, remote health monitoring systems have been dramatically growing over the last few decades. Such systems enable collecting health and well-being data continuously and real-time. Nowadays, modern wearable devices (e.g., smartwatches and smart rings) are widely used to collect biomedical signals, such as Photoplethysmogram and Electrocardiogram (King & Sarrafzadeh, 2018).

Photoplethysmography (PPG) is an optical method to measure changes in blood volume in tissues. The device is placed on the skin, conventionally on a person's fingertip (pulse oximetry) or wrist (smartwatch) (Castaneda, Esparza, Ghamari, Soltanpur, &

* Corresponding author.
E-mail address: mohammad.feli@utu.fi (M. Feli).

https://doi.org/10.1016/j.smhl.2023.100390
Received 14 March 2023; Accepted 15 March 2023
Available online 20 March 2023



(a) Reliable PPG (b) Unreliable PPG

Fig. 1. PPG Samples. (a) The signal is clean. (b) The signal is distorted due to motion artifacts.

Nazeran, 2018). Then, the PPG signal is recorded by exposing light with a particular wavelength (e.g., infrared, red, and green) to the skin and capturing the reflected or transmitted light using photodetectors. PPG has become increasingly popular for measuring multiple vital signs, such as heart rate, heart rate variability, and oxygen saturation (SpO_2). The method is, however, highly vulnerable to artifacts caused by, for example, user's hand movement or environmental noise (Orphanidou, 2017). Therefore, it is still a major challenge in real-world applications to collect noise-free PPG during daily routine activities.

Noise and artifacts reduce the signal quality, distort the signal, and pose significant challenges to PPG analysis. Fig. 1 demonstrates examples of clean ("Reliable") and corrupted ("Unreliable") PPG. When the signal quality is poor, vital signs cannot be extracted reliably, leading to errors in decision-making. In addition, noisy data collection and transmission leads to a waste of energy in remote health monitoring. According to earlier studies (Deepu et al., 2016; Reddy et al., 2022), data transmission is the major source of energy consumption (up to 90%) in health monitoring devices. Therefore, it is essential to assess PPG signal quality at the network's edge (e.g., sensors or gateway devices) to reduce energy consumption, as well as prevent erroneous decision-making and life-threatening consequences.

To address the noisy PPG challenge, PPG signal quality assessment (SQA) methods have been introduced in the literature. In some studies (Li et al., 2011; Liu et al., 2010; Reddy et al., 2020; Vadrevu & Manikandan, 2019), decision rules have been used to distinguish between clean and corrupted signals. Most of these rules have been defined based on physiological features and signal characteristics. Although these approaches are fast and easy to implement, they have low accuracy and poor generalization to unseen artifacts. Conventional machine learning (ML) methods have also been employed for PPG SQA (Chong et al., 2014; Li & Clifford, 2012; Mahmoudzadeh, Azimi, Rahmani, & Liljeberg, 2021; Pereira et al., 2019; Roy et al., 2020). However, most ML-based PPG SQA techniques have been developed only with a focus on classification accuracy, and they have not been evaluated in terms of execution time and energy consumption. Moreover, deep learning techniques have been proposed in some studies (Azar, Makhoul, Couturier, & Demerjian, 2021; Liu, Li, Wang, Chen, & Su, 2020; Naeini, Azimi, Rahmani, Liljeberg, & Dutt, 2019; Shin, 2022) to ensure a high degree of accuracy. Despite the fact that deep learning models do not require handcrafted feature extraction, their high cost makes them infeasible for real-time analysis. In addition, existing PPG SQA methods (Pereira et al., 2019; Shin, 2022; Vadrevu & Manikandan, 2019) aim to reduce false negative prediction, which indicates false alarms, while false positive prediction needs to be considered as well.

A PPG SQA method should be accurate, energy efficient, and fast enough to perform efficiently on low-power devices. It is, therefore, essential to consider execution time and energy consumption in developing a PPG SQA technique. To this end, the PPG SQA methods should be evaluated on different low-power devices. Moreover, we believe reducing both false positive rate and false negative rate are critical in developing a PPG SQA model. False positive prediction leads to unreliable PPG data collection, wasting a large amount of energy and causing erroneous decision-making. In contrast, false negative prediction results in false alarms in health monitoring systems.

In this paper, we propose a lightweight semi-supervised PPG quality assessment method for real-time PPG-based health monitoring systems. First, we extract a wide range of features from different domains of the signals and investigate them regarding the lowest execution time and highest discriminating power. The best features are selected using a feature selection method. Then, we develop a one-class support vector machine model to classify signals into "Reliable" and "Unreliable" classes. Finally, we evaluate the proposed method using a PPG dataset collected via smartwatches from 46 individuals in a remote health monitoring study. The proposed method is compared with a rule-based, two traditional machine learning, and two deep learning PPG SQA methods. The methods are evaluated in terms of accuracy, execution time, and energy consumption on two embedded devices. In summary, the contributions of this paper include the following:

1. Proposing a lightweight and energy-efficient PPG SQA method applicable to low-power devices.
2. Extracting comprehensive features from different domains of the PPG signals and investigating them to select the best features in terms of accuracy and execution time.
3. Developing a one-class SVM model based on the strong similarity between "Reliable" PPG signals.
4. Evaluating the proposed method using PPG data collected from 46 persons via smartwatches in free-living conditions.
5. Comparing the proposed method with five state-of-the-art PPG SQA methods in terms of accuracy, execution time, and energy consumption.

The outline of the rest of the paper is as follows. First, we discuss related works in Section 2. Next, Section 3 provides a brief overview of the background. Section 4 outlines the PPG dataset used in this study. Then, the proposed method is described in Section 5. Implementation and evaluation details are presented in Section 6. Finally, Section 7 concludes the paper.

2. Related works

In this section, we present the state-of-the-art PPG quality assessment methods proposed in the literature. We divide the methods into three main groups: rule-based, traditional machine learning, and deep learning methods.

Rule-based techniques extract several features from PPG and then use a set of thresholds to distinguish clean signals from corrupted ones (Li et al., 2011; Liu et al., 2010). Vadrevu and Manikandan (2019) proposed a rule-based PPG quality assessment method enabled by absolute amplitude, zero-crossing rate, and autocorrelation features. Another rule-based approach was presented by Reddy et al. (2020) to reduce false alarms through the detection of PPG sensor disconnection and signal saturation. Although rule-based methods are fast and easy to implement, they lack generalization to unseen data because their decision rules are defined based on expected signal characteristics, and their thresholds are set empirically. In other words, they have been developed based on their own data specifications, resulting in lower accuracy when faced with new data.

In addition, traditional supervised machine learning algorithms have been exploited for PPG SQA. Li and Clifford (2012) extracted various features using dynamic time warping (DTW) and then fed the features to a multi-layer perceptron neural network to classify beats into good and bad quality classes. However, the main drawback of DTW is its high time complexity, making it challenging to use in real-time applications. In Chong et al. (2014), a support vector machine (SVM) was trained using time-domain features, such as the standard deviation of peak-to-peak intervals and amplitude, the standard deviation of systolic and diastolic interval ratios, and mean standard deviation of pulse shape. Pereira et al. (2019) also deployed three supervised algorithms, including k-nearest neighbors, decision trees, and two-class SVM. They extracted forty-two time- and frequency-domain features from the 30-seconds PPG signals. For feature extraction, most supervised PPG SQA methods only focus on the discrimination capability (accuracy) without considering execution time. However, extracting some of the proposed features is cumbersome and leads to high latency in real-time analysis. Furthermore, supervised methods require a large amount of labeled data from both “Reliable” and “Unreliable” classes. It is, however, expensive and time-consuming to annotate massive data manually.

Unsupervised machine learning approaches have also been employed in PPG SQA methods in the literature. In Roy et al. (2020), a few entropy and signal complexity-related features were extracted from PPG and then fed to a self-organizing map (SOM) model to perform SQA. In another study by Mahmoudzadeh et al. (2021), the elliptic envelope, an unsupervised anomaly detection method, was developed for lightweight PPG SQA. Although unsupervised PPG SQA methods do not require data annotation phase and are fast to implement, their classification precision is mostly lower than supervised methods.

Recent studies have proposed the use of deep learning techniques in PPG SQA (Naeini et al., 2019; Shin, 2022). Liu et al. (2020) segmented PPG and differential PPG signals into heart cycles, transformed them into 2D images, and then fed the images to two-dimensional deep convolution neural network (CNN) models to perform SQA. They utilized two renowned deep learning architectures for classification: ResNet-50 (He, Zhang, Ren, & Sun, 2016) and VGG-19 (Simonyan & Zisserman, 2014). In another study, a deep autoencoder was employed to perform PPG SQA (Azar et al., 2021). To train the autoencoder, only good-quality samples were used to learn the normal pattern of data, and then input samples that did not follow that pattern were classified as bad-quality signals. However, the accuracy of the method is highly dependent on the reconstruction error threshold used to distinguish “Reliable” and “Unreliable” signals. Furthermore, using deep learning models in real-time analysis (e.g., at the edge) results in high latency and energy consumption.

Despite the significant efforts made in previous studies, two challenges still need to be overcome. We describe these issues as follows:

- The major challenge in existing PPG SQA techniques is their high execution time and power consumption. The SQA model should be as fast as possible to be applicable to low-energy wearable devices. To achieve this, evaluating the PPG SQA approach on real low-energy devices is essential. However, most current techniques have not been proposed for real-time analysis. Even though a few studies conducted real-time experiments (Reddy et al., 2020; Vadrevu & Manikandan, 2019), there is still room for improvement.
- Another issue with existing PPG SQA methods is that they ignore the importance of the false positive rate (FPR). In PPG SQA models, FPR denotes the probability of misclassifying “Unreliable” samples as “Reliable”. On the other hand, the probability of misclassifying “Reliable” signals as “Unreliable” is indicated by the false negative rate (FNR). To the best of our knowledge, FPR is more critical than FNR because FPR leads to unreliable data collection, life-threatening consequences, and massive energy wastage; however, FNR only results in false alarms in health monitoring systems. Therefore, an effective PPG SQA method should provide the lowest FPR.

3. Background

3.1. Photoplethysmogram (PPG)

Photoplethysmography is a non-invasive optical technique that measures the blood volume changes on the surface of perfused tissues, such as the fingertip, wrist, ear-lobe, and forehead (Kyriacou & Allen, 2021). PPG signals can be recorded via transmission or

Table 1
Participants' background information.

Characteristic	Type	Values
Age, mean (SD)	Men	33 (6)
	Women	31.5 (6.6)
BMI, mean (SD)	Men	24.4 (5.6)
	Women	25.5 (2.9)
Physical activity, n (%)	Almost daily	12 (27)
	Once a week	9 (20)
	>Once a week	21 (47)
Employment status, n(%)	Working	32 (71)
	Unemployed	1 (2)
	Student	8 (18)
	Other	1 (2)

reflection methods. In the transmission method, a light-emitting diode (LED) emits light to the tissue, and a photodetector captures the intensity of non-absorbed light on the other side of the tissue. On the other hand, both LED and photodetector are placed on the same side of the tissue in the reflection method (Castaneda et al., 2018).

PPG signals consist of slowly varying baseline (DC) and pulsatile (AC) components. The former varies due to the respiratory cycle and the amount of light absorbed by, for example, tissue and bones. The AC component indicates the pulsation oscillated with the cardiac cycle, including the systolic and diastolic stages of the heart activity. Therefore, the PPG signal can be used for heart rate (HR) and heart rate variability (HRV) extraction applications. A view of a PPG sample has been shown in Fig. 1(a).

PPG is an easy-to-implement method used in various wearable devices. However, reliable PPG collection is challenging, particularly in home-based applications, since the method is highly susceptible to noise (Kyriacou & Allen, 2021). These include motion artifacts, low- and high-frequency noise, and baseline wander. Moreover, The signal quality is highly vulnerable to the sensor type and measurement site location. The noise can be reduced in medical settings by considering an appropriate data acquisition protocol and minimizing the subject's movements. However, high-quality signals cannot be obtained when the data is collected with wearable sensors in real-world ambulatory environments.

3.2. Semi-supervised one-class learning

Semi-supervised learning is a group of ML methods, which involve portions of labeled and unlabeled data. In this approach, an initial model is trained with a small number of labeled samples, and then a large number of unlabeled samples are applied iteratively to the model (Van Engelen & Hoos, 2020). Semi-supervised learning bridges the gap between supervised and unsupervised learning methods, in which only labeled or unlabeled data are used for training. Semi-supervised learning has been widely used in the literature for anomaly detection, for example, when normal samples are present, but numerous unseen types of anomalies are not discovered (Villa-Pérez et al., 2021).

One-class learning is a semi-supervised approach in which solely data from one class is used for training (Schölkopf et al., 1999). This method is commonly used in classification problems where only data of one class is available, but the other class is unseen, or the behavior of its samples is unknown. For example, in novelty detection, the one-class model is constructed with normal samples and then tested with both normal and novel samples (Domingues, Filippone, Michiardi, & Zouaoui, 2018). In contrast to two-class learning methods – which separate two classes with a decision boundary – in one-class approaches, the discriminant function creates the hyperplane decision surface based on only one class.

4. Dataset

We evaluate our proposed method using a PPG dataset collected in a remote health monitoring study (Mehrabadi et al., 2020). In this study, the participants were asked to continuously wear a smartwatch – Samsung Gear Sport watch (0000) – to capture their vital signs, physical activity, and sleep parameters. The health data were collected 24 h from individuals engaging in everyday routines and activities throughout the monitoring period.

The recruitment was performed in southern Finland from July to August 2019. The data were objectively collected from 46 participants (23 women and 23 men). Diversity in individuals regarding age, weight, physical condition, education level, and lifestyle was taken into account. In addition, different criteria were considered, such as the restriction of wearing watches at work and physical exercises. All participants were in good health with no cardiovascular diseases and aged between 18 and 55. The study's aim was presented to the candidates in face-to-face meetings. Table 1 summarizes the background information of the participants. The table includes 42 participants since the background information of 4 participants is unavailable.

Samsung Gear Sport watch (0000) was used to collect PPG signals. The watch is light and waterproof and is convenient for long-term data collection during various activities. Its size and weight (including the strap) are $44.6 \times 42.9 \times 11.6$ mm and 67 g, respectively. The watch – equipped with optical and inertial measurement unit (IMU) sensors – run the open-source Tizen operating system, which enabled us to develop data collection and analysis applications. The optical sensors record PPG signals at a sampling frequency of 20 Hz.

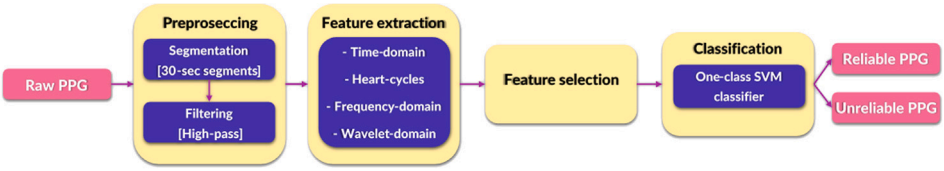


Fig. 2. Pipeline of the proposed PPG SQA method in the design time phase.

The participants were instructed to wear the watch on their non-dominant hand continuously. Additionally, an application was developed to transmit PPG signals from the watch to our cloud server via Wi-Fi. Sixteen-minute PPG signals were recorded every thirty minutes. Due to the sensor calibration, we discarded the first and last minutes of each PPG record in our analysis. Note that there was no need to charge the smartwatch during the one-day experiment due to the watch's battery capacity.

Ethics: This study was conducted according to the ethical principles based on the Declaration of Helsinki and the Finnish Medical Research Act (#488/1999). The study protocol received a favorable statement from the ethics committee (University of Turku, Ethics committee for Human Sciences, Statement #44/2019). The participants were informed about the study, both orally and in writing, before obtaining their consent. Participation was voluntary, and all participants had the right to withdraw from the study at any time and without giving any reason.

The PPG dataset contains a large number of noisy signals, which poses a significant challenge to analysis. The noise is nonstationary and varies throughout the monitoring due to the user's hand movement and body motion. For this reason, the daytime signals are more distorted – due to noise and artifacts – than the nighttime signals. In general, the dataset includes 155.5 h of "Reliable" PPG signals and 387.4 h of "Unreliable" PPGs.

5. Proposed PPG quality assessment

In this section, we propose an ML-based method for real-time and energy-efficient PPG SQA. The aim is to develop an accurate and lightweight method to perform PPG SQA on low-power devices. Wearable sensors collecting PPG signals include limited battery and computational resources. Therefore, they require lightweight methods with minimum execution time and energy consumption. To this end, we develop a lightweight PPG SQA technique and evaluate its performance using embedded devices.

Our proposed method includes four main components in the design time phase (see Fig. 2), through which "Reliable" and "Unreliable" PPG signals are differentiated. First, preprocessing techniques are employed to segment the signals and discard frequencies outside the heartbeat frequency range. Then, various features are extracted from the filtered PPG. In this study, we investigate several time- and frequency-domain features. Then, the best features, in terms of accuracy and latency, are selected to be used in the proposed method. Finally, we develop a one-class SVM method to perform the signal quality assessment. The classification model is built by feeding the selected features to the model. In the following, we describe the different components of the proposed method in more detail.

5.1. Preprocessing

The PPG signal preprocessing in our analysis includes segmentation and filtering. We first divide PPG signals into quasi-stationary segments. The PPG signals collected by the wearable devices are non-stationary in terms of noise levels. For example, our dataset includes 15-minute records in which the noise levels vary over time. For the analysis, we divide the signals into 30-second non-overlapping segments. We assume that the 30-second segments are "Reliable" (i.e., noise-free) or "Unreliable" (i.e., including false peaks). It should be noted that the length of the segment should be long enough to provide meaningful time and frequency information for the analysis (i.e., feature extraction and classification), and should be short enough to ensure the noise level is stationary (Mahmoudzadeh et al., 2021; Pereira et al., 2019). We then filter the segments to mitigate the baseline wander. To this end, we employ a second-order Butterworth highpass filter with a cut-off frequency of 0.5 Hz.

Annotation: We manually annotate PPG segments as "Reliable" and "Unreliable" samples. The annotation is performed according to the morphology of the waveforms (i.e., the visibility of the systolic peak). For this purpose, the (30-second) segments are labeled as "Reliable" if cardiac cycles are clear (i.e., no false peaks). Otherwise, the segments are labeled as "Unreliable". After annotation, the labeled signals are rechecked to minimize mislabeling.

5.2. Feature extraction

In this study, we extract forty-five features from the PPG segments, divided into four categories: i.e., time-domain, heart-cycles, frequency-domain, and wavelet-domain of the signal. The features have been listed in Table 2 and are briefly outlined as follows:

Time-domain of the signal: We derive time-domain features, showing the signal's morphology variation over time. Motion artifacts considerably affect the PPG signals, distorting cardiac waveforms. Therefore, time-domain features might be beneficial in distinguishing good and bad-quality signals. Entropy measures can also be computed to indicate the signals' regularity, complexity, and unpredictability in time. The time-domain features extracted from the signals in this study are listed in Table 2.

Table 2
Extracted features from PPG signals.

Category	Feature	Description
Time-domain of the signal	STD[a], skewness, and kurtosis of the signal	Measures of dispersion, asymmetry, and tailedness
	Permutation entropy of the signal	Measure of complexity
	Approximate entropy of the signal	Measure of regularity and unpredictability of fluctuations
	Sample entropy of the signal	Measure of complexity
	SVD[b] entropy of the signal	Measure of richness
	Hjorth parameters of the signal	Measure of mobility and complexity
	STD of first derivative of the signal	Measures of dispersion of changes in adjacent points
	Zero crossing rate of the signal	The number of times the signal crosses the horizontal axis
	Interquartile range of the signal	Measure of dispersion of middle 50% of the data
	MAP[c] of the signal	Identifying low amplitude to detect sensor disconnection
Heart-cycles of the signal	STD, skewness, and kurtosis of HCs[d]	Measures of dispersion, asymmetry, and tailedness of HCs
	Energy of HCs	The area under the squared magnitude of HCs
	SVD entropy of HCs	Measure of richness of HCs
	Permutation entropy of HCs	Measure of complexity of HCs
	Approximate entropy of HCs	Measure of regularity and unpredictability of HCs
	Sample entropy of HCs	Measure of complexity of HCs
	Spectral entropy of HCs	Measure of spectral power distribution in HCs
	Correlations between a template and HCs	Measure of dependency between HCs in a PPG signal
	Euclidean distances between a template and HCs	Measure of similarity between HCs in a PPG signal
Frequency-domain of the signal	Kurtosis of Fourier spectrum of the signal	Measure of asymmetry of Fourier domain of the signal
	STD of power spectral density of the signal	Measure of dispersion of signal's power content versus frequency
	Spectral entropy	Measure of spectral power distribution
Wavelet-domain of the signal	Skewness and kurtosis of wavelet domain of the signal	Measures of asymmetry and tailedness of the approximation coefficients of wavelet transform
	Permutation entropy of wavelet domain of the signal	Measure of complexity of the approximation coefficients of wavelet transform
	SVD entropy of wavelet domain of the signal	Measure of richness of approximation coefficients of wavelet transform
	Approximate entropy of wavelet domain of the signal	Measure of regularity and unpredictability of approximation coefficients of wavelet transform
	Hjorth parameters of wavelet domain of the signal	Measure of mobility and complexity of approximation coefficients of wavelet transform

[a]Standard deviation.
[b]Singular value decomposition.
[c]Maximum absolute amplitude.
[d]Heart-cycles.

Heart-cycles of the signal: We extract features from the cardiac cycles in PPG segments. It is evident that noise distorts the regular shape of the pulse waves (see Fig. 1). In other words, unlike clean PPG, the shape of the heart cycles varies in a corrupted PPG. Thus, we can assess the PPG signal quality by investigating the variation of the cardiac waveforms within the PPG segments. To extract heart cycle features, we first detect systolic peaks in a segment (Elgendi, Norton, Brearley, Abbott, & Schuurmans, 2013). Then, a cardiac cycle is obtained using a fixed time interval, the center of which is the corresponding systolic peak. After extracting the heart cycle features, the standard deviation and range of each feature are calculated. The range is calculated by subtracting each feature's minimum value from the maximum value.

In addition, we use the template matching technique to extract similarity features from heart cycles. The template matching approach investigates the signal's regularity by measuring the similarity between a template and signal windows. A clean PPG is quasi-periodic, as it shows the rhythmic activity of the cardiac system. Accordingly, there is a high degree of similarity between a signal's template and heart cycles in clean PPGs. In contrast, this similarity is low for corrupted PPGs. In our analysis, we select the average of segment's cardiac cycles as the template. We then calculate Euclidean distances and correlations between the template and each cardiac cycle. Subsequently, the average, standard deviation, and Shannon entropy of the distances and correlations are obtained. The heart cycle features are listed in Table 2.

Frequency-domain features: We use the Fourier transform to investigate the presence of different frequencies within PPG segments. The dominant or resonance frequency in a "Reliable" PPG signal lies within a specific range between 0.5 Hz and 3.3 Hz based on a person's heart rate range, which varies between 30 and 200 beats per minute (Moraes et al., 2018). Motion artifacts distort the signal's frequency distribution and dominant frequency. We extract various features (listed in Table 2) showing the distribution of the signal in the frequency domain.

Wavelet-domain features: We exploit the wavelet transform to decompose the PPG segments and then extract features from low- and high-frequency components. The method provides both temporal and spectral information of the signal. As mentioned previously, noise corrupts both time and frequency components in PPG signals. Therefore, analyzing the decomposed components can be leveraged to assess the signal quality. In this study, we extract wavelet features from the wavelet approximation coefficients of the PPG segments (see Table 2).

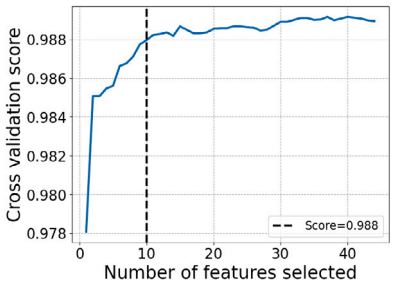


Fig. 3. Cross validation scores obtained by RFE method.

Table 3
RFE rank and execution time of the features.

Feature	RFE rank	Time (ms)
Range of energy of HCs[a]	**1**	**2.27**
Average Euclidean distances between a template and HCs	**2**	**2.41**
Average Correlations between a template and HCs	**3**	**2.49**
STD[b] of permutation entropy of HCs	4	3.27
Range of spectral entropy of HCs	5	6.25
STD of energy of HCs	6	2.24
STD of Correlations between a template and HCs	7	2.41
STD of power spectral density of the signal	**8**	**0.23**
Interquartile range of the signal	**9**	**0.18**
Range of skewness of HCS	10	4.12

[a]Heart-cycles.
[b]Standard deviation.

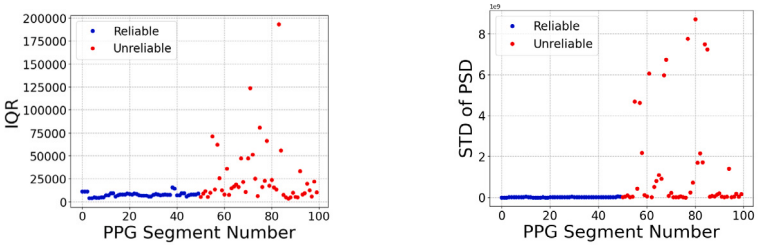
5.3. Feature selection

In this section, we select a few features among all extracted features based on accuracy and execution time. Extracting all the features mentioned in the previous subsection is time-consuming in real-time analysis. Therefore, we select only features with the lowest latency and highest discriminating power. For this purpose, we first use a Recursive Feature Elimination (RFE) method (Guyon, Weston, Barnhill, & Vapnik, 2002) to find the best features in terms of accuracy. In this technique, a support vector model is given to RFE as an estimator that assigns weights to features. RFE starts with training the estimator on the initial features, obtaining the importance of each feature, and removing the less important features. This procedure is repeated recursively until the algorithm achieves the desired number of features.

We leverage RFE with k-fold cross-validation on the PPG dataset to tune the number of selected features. In cross-validation feature selection, the feature set is split into k subsets. Then, the model is validated k times, where k-1 subsets of the feature set are used for training each time. We apply cross-validation with k = 5 to find the optimum number of the selected features in RFE. Fig. 3 shows cross-validation scores for different numbers of features. As indicated, the cross-validation score increases significantly by selecting the top 10 features and then improves gradually, including the other features.

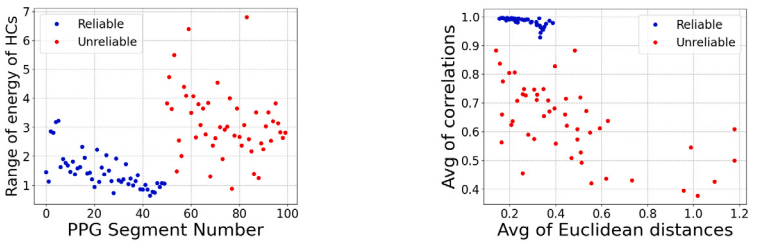
Moreover, we calculate the execution time of the top 10 features. The execution time is the amount of time used to extract a feature from the signal. We select the best features based on both RFE ranking and latency. Table 3 presents the top 10 features with the highest RFE ranking and their execution time for 30-second PPG segments. We select three high-ranked features with low execution time: i.e., (1) the range of energy of heart cycles, (2) the average of Euclidean distances between a template and heart cycles, and (3) the average of their correlations. Additionally, we select the standard deviation of power spectral density and interquartile range of the signal due to their low execution time. Therefore, we select five features for the PPG analysis to be used in the runtime phase of the proposed method. The selected features will be described as follows.

- **Interquartile Range of the Signal**
  Interquartile range (IQR) is a statistical feature that measures the spread of the middle half of the data distribution. It is defined as the signal's first quartile value (Q1) from the third quartile (Q3). We obtain the IQR of the PPG segments to assess the variability of the central portion of the data. The IQR values of "Unreliable" PPG signals are high and dispersed due to the irregularity of the signals. However, IQR values are low for "Reliable" signals and lie within a specific range. Fig. 4(a) illustrates IQR values of 100 "Reliable" and "Unreliable" PPG segments. As shown, the IQR of "Unreliable" PPG signals is more dispersed and higher than that of "Reliable" signals.
- **STD of Power Spectral Density of the Signal**
  Power spectral density (PSD) shows the distribution of the power in the PPG signal as a function of frequency. Fig. 5 demonstrates the PSD of a "Reliable" and an "Unreliable" PPG signal. PSD of the "Reliable" PPG contains the highest signal



(a) IQR of the signals

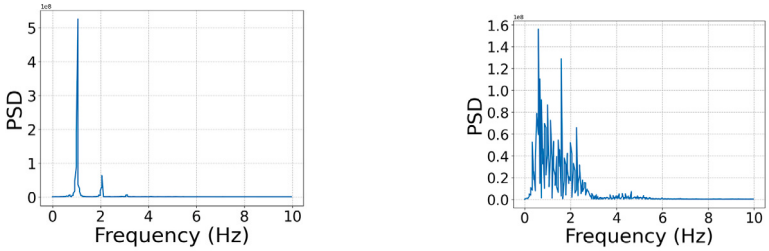
(b) STD of PSD of the signals



(c) Range of energy of heart cycles

(d) Template matching features

Fig. 4. Difference between Reliable (blue) and Unreliable (red) samples using selected features.



(a) Reliable PPG

(b) Unreliable PPG

Fig. 5. Power spectral density of two PPG segments.

power at the first peak that corresponds to the heartbeat frequency. However, in "Unreliable" PPG, noise and artifacts alter the PSD, where the power is high over a wide range of frequencies. Fig. 4(b) illustrates the standard deviation of PSD of 100 "Reliable" and "Unreliable" PPG segments. It is obvious that the PSD of "Unreliable" segments is more dispersed than "Reliable" segments.

We obtain PSD of PPG using Welch's method (Welch, 1967). Welch's method computes the PSD of a signal x by dividing it into M successive segments and calculating the discrete Fourier transform of each segment. Then, the squared magnitude of the result is computed and averaged. The PSD is calculated as the following equation.

$$S_{xx} = \frac{1}{M}\sum_{m=0}^{M-1} |DFT(x_m)|^2 \triangleq \{|X_m(\omega_k)|^2\}_m \tag{1}$$

- **Range of Energy of Heart Cycles**

The energy of a heart cycle is defined as the area under the squared magnitude of a heart cycle. It is obtained as follows.

$$E_{hc} = \sum_{n=1}^{N} |x_{hc}(n)|^2 \tag{2}$$

where x_{hc} is the heart cycle and N is the number of heart cycle samples.

Artifacts and noise (e.g., signal saturation and sensor disconnection) distort the regular shape of the PPG, modifying the signal's energy. In addition, nonstationary noises vary the signal energy across heart cycles. The energy values of heart cycles are almost similar across "Reliable" PPGs. However, they vary in "Unreliable" PPGs. Therefore, the range of energy values of cardiac cycles can be a good indicator for PPG SQA. Fig. 4(c) shows the range of energy of heart cycles for 100 "Reliable" and "Unreliable" PPGs. It is indicated that "Reliable" PPG heart cycles have a lower energy variation than "Unreliable" PPGs.

- **Average Euclidean Distances between a Template and Heart Cycles**

We exploit a template matching approach to measure the similarity between cardiac cycles. In "Reliable" PPGs, there is a high degree of similarity between cardiac cycles; however, this similarity is low in "Unreliable" PPGs. We consider the averaged cardiac cycles as the template. Then, we measure the Euclidean distances between the template and each cardiac cycle (as Eq. (3)). Finally, we calculate the average distances for each PPG segment.

$$d(template, hc_i) = \sqrt{\sum_{j=1}^{l}(hc_{i,j} - template_j)} \tag{3}$$

where l is the length of a heart cycle, and d is the Euclidean distance between the template and a heart cycle.

- **Average Correlations between a Template and Heart Cycles**
  In addition to the Euclidean distances, we calculate the correlation coefficients in template matching. The correlation between heart cycles in "Reliable" PPGs is high, while it is low in "Unreliable" PPGs. We compute the Pearson's correlation coefficients between the template and each heart cycle and then average it over all heart cycles in the signal. Pearson's correlation coefficients are obtained as follows.

$$r = \frac{\sum_{i=1}^{l}(hc_i - \hat{hc})(template_i - \hat{template})}{\sqrt{\sum_{i=1}^{l}(hc_i - \hat{hc})^2 \sum_{i=1}^{l}(template_i - \hat{template})^2}} \tag{4}$$

where $\hat{hc}$ and $\hat{template}$ are the mean of the heart cycle and template, respectively.

Fig. 4(d) illustrates the average of Euclidean distances and correlations in template matching for 100 "Reliable" and "Unreliable" PPGs. As shown, cardiac cycles in "Reliable" signals have low Euclidean distances and high correlations, but they are dispersed in "Unreliable" PPGs.

5.4. Classification

We propose an ML method to classify "Reliable" and "Unreliable" PPG signals. The model receives the features extracted from a PPG signal as input and predicts the signal's class label as the output: i.e., the quality of the PPG. Our proposed PPG SQA is designed by considering the difference between the dispersion of "Reliable" and "Unreliable" PPG signals: i.e., "Reliable" PPG samples are similar while "Unreliable" samples are highly dispersed.

To visualize this difference, we use the Principal Component Analysis (PCA) method (Jolliffe & Cadima, 2016), which reduces the data dimension (i.e., the five features extracted from the signal) to two principal components. Fig. 6 shows the dispersion of two principal components for the "Reliable" and "Unreliable" PPG samples. As indicated, "Reliable" instances are clustered densely, while "Unreliable" instances are widely dispersed. The dispersion and unpredictability of the "Unreliable" PPG samples are due to the variety of non-stationary noises, including hand movements, body motions, and environmental noises.

To this end, we develop a semi-supervised one-class SVM (OCSVM) method for PPG SQA. OCSVM was introduced for the first time for novelty detection by Schölkopf et al. (1999). This method was designed based on the SVM algorithm with only one training class. Similar to conventional SVM, OCSVM maps the data into a feature space using a kernel function to separate them with the maximum margin. However, the OCSVM model is trained only using one class, as opposed to conventional two-class SVM models.

Our proposed OCSVM model is trained solely by "Reliable" samples and then tested by both "Reliable" and "Unreliable" samples. The trained OCSVM model encompasses all the "Reliable" samples (training instances) using a hypersphere, and then the inference classifies new samples (test instances) based on their position from the hypersphere. Samples inside the hypersphere are classified as "Reliable"; otherwise, they are classified as "Unreliable". The proposed OCSVM model aims to find the maximal margin for the hypersphere. The OCSVM discriminant function is obtained as Eq. (5), which produces a positive value for the region that contains "Reliable" samples and a negative value for the "Unreliable" region.

$$f(x) = sign\left(\sum_i \alpha_i k(x_i, x) - \rho\right) \tag{5}$$

x_i is a support vector, and k denotes the kernel function: i.e., the polynomial kernel function in our analysis:

$$k(x_1, x_2) = (a + x_1^T x_2^T)^b \tag{6}$$

where a is a constant term, and b represents the degree of the kernel. α_i – in Eq. (5) – is obtained with the following dual expression:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\sum_{ij}\alpha_i\alpha_j k(x_i, x_j) \\ s.t \quad &0 \le \alpha_i \le \frac{1}{vl}, \qquad \sum_{\alpha_i} = 1 \end{aligned} \tag{7}$$

where $v \in (0, 1)$ and l is the number of samples. ρ – in Eq. (5) – is also calculated based on the Karush-Kuhn–Tucker (KKT) condition (Boyd & Vandenberghe, 2004):

$$\rho = \sum_j \alpha_j k(x_j, x_i) \tag{8}$$

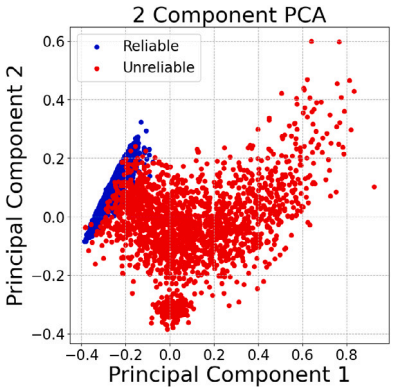
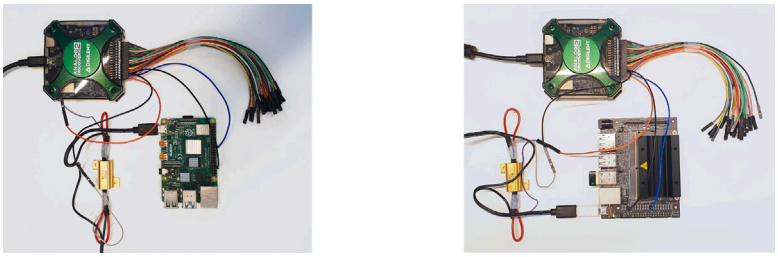


Fig. 6. Dispersion of Reliable (blue) and Unreliable (red) samples in the first and second PCA components.



(a) Raspberry Pi 4 (b) NVIDIA Jetson Nano

Fig. 7. Measurement setups for two embedded boards.

6. Implementation and evaluation

We implement and evaluate the proposed method in terms of accuracy, execution time, and energy consumption. In the following, we first outline the experimental setups. Then, we briefly describe existing PPG SQA methods (as baseline methods), with which our method is compared. Finally, the experimental results are presented.

6.1. Experimental setups

To evaluate the accuracy, execution time, and energy consumption of the methods, we use two embedded single-board computers: i.e., a Raspberry Pi 4 (Gay, 2014) operated by a 64-bit Debian Linux OS and an NVIDIA Jetson Nano (Plawrence, 2019) operated by a 64-bit Ubuntu Linux OS. Table 4 shows the specification of the computer boards.

We install Docker (Merkel, 2014), an OS-level virtualization tool, on both devices and prepare a Docker image containing Python libraries necessary for running the proposed method and all the baseline methods codes (e.g., Scipy Virtanen et al., 2020, TensorFlow Abadi et al., 2015, and Neurokit2 Makowski et al., 2021). Even though we run the code on different devices with different operating systems, using Docker technology ensures that the running environment is exactly the same in terms of the Python version, the list of installed libraries, and their versions.

The energy consumption setup consists of a shunt resistor and a *Digilent Analog Discovery 2* digital oscilloscope. We put a 25 W 0.01 Ω resistor on the power line serial to the device and record the amount of voltage dropped over the resistor to calculate the current change and, consequently, the energy consumption. While the oscilloscope's first channel is measuring the voltage, the second channel is connected to the board's GPIO 24 to mark the beginning and the end of code execution. Within the code, we put GPIO enabling/disabling commands at the beginning and the end of the calculation, excluding the part that reads data from the disk into the memory. Turning GPIO 24 On and Off appears as a vertical line on channel 2 and marks the beginning and end of the code execution on the channel 1 signal. We record both channels' signals during the code execution with a 200 Hz sampling rate. We measure the average of the channel 1 recorded signal during the code execution (using the marks recorded in channel 2) and the average during the system idle. Subtracting these two values shows the exact effect of the executed code on increasing the current consumption that leads to dropping the voltage over the shunt resistor. We monitor both boards without peripherals (i.e., keyboard, mouse, display) over SSH communication enabled via WiFi. Fig. 7 shows the setup for both computer boards, and Fig. 8 illustrates the oscilloscope screen at the end of a measurement.

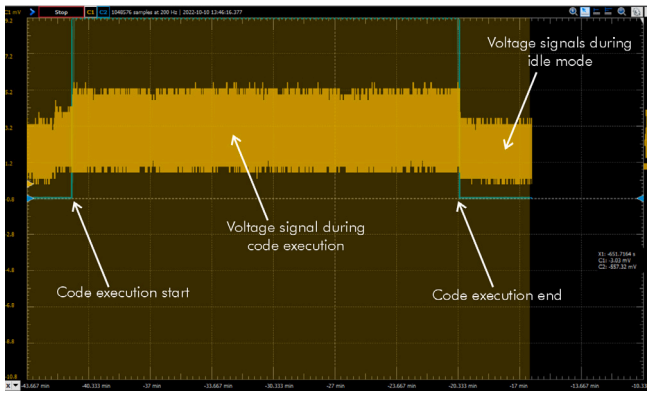


Fig. 8. Oscilloscope screen at the end of a measurement.

Table 4
Embedded boards specifications.

	Raspberry Pi 4	NVIDIA Jetson Nano
CPU	ARM Cortex-A72	ARM Cortex-A57
CPU architecture	ARM v8	ARM v8
CPU type	64 bit	64 bit
CPU frequency	600 MHz–1.5 GHz	102 MHz–1.43 GHz
CPU cores	4	4
L1 cache	32 kB	32 kB
L2 cache	1 MB	2 MB
RAM	4 GB	4 GB
RAM type	LPDDR4	LPDDR4
RAM speed	1.4 GB/s	3.2 GB/s
GPU architecture	VideoCore VI 3D	Maxwell
GPU cores	0	128
Storage	32 GB, external MicroSD card	32 GB, external MicroSD card
GPIO	40 pins	40 pins
Thermal dissipation	3 small heat sinks installed on all SoCs	One big heat sinks installed on the SoC
Power source	5V DC 3A	5V DC 3A
OS	Linux Debian 11 - aarch64 GNU/Linux	Ubuntu 18.04.6 LTS - aarch64 GNU/Linux

Table 5
Baseline methods.

Authors	Category	Method	Year
Vadrevu and Manikandan (2019)	Rule-based	Maximum absolute amplitude, Zerocrossing rate, Autocorrelation	2019
Mahmoudzadeh et al. (2021)	Unsupervised	Elliptical Envelope	2021
Pereira et al. (2019)	Supervised	Time- and Frequency-domain features + SVM	2020
Naeini et al. (2019)	Deep learning	Single layer 1D-CNN	2019
Shin (2022)	Deep learning	Multi-Layer 1D-CNN	2022

6.2. PPG SQA baseline methods

We compare our proposed method with five baseline PPG SQA methods. The baseline methods were selected from different PPG SQA categories (i.e., rule-based, unsupervised, supervised, and deep learning techniques). These methods are listed in Table 5. The rule-based technique (Vadrevu & Manikandan, 2019) contains decision rules enabled by statistical features, such as maximum absolute amplitude, zero-crossing rate, and autocorrelation. In the unsupervised method (Mahmoudzadeh et al., 2021), five statistical and entropy features are used to train an Elliptical Envelope model. The supervised approach (Pereira et al., 2019) utilizes a wide range of time- and frequency-domain features to train an SVM classifier. Two deep learning techniques (Naeini et al., 2019; Shin, 2022), enabled by single-layer and multi-layer CNN architectures, are also selected.

6.3. Results

In this subsection, we present the experimental results of the proposed and baseline methods in terms of accuracy, execution time, and energy consumption.

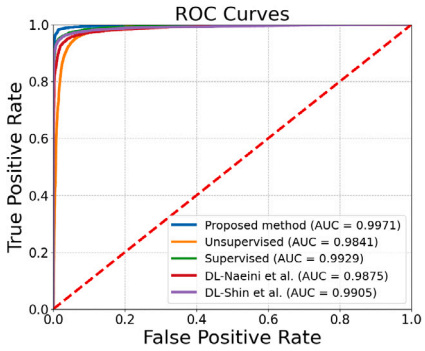


Fig. 9. ROC curves of the proposed and ML methods.

6.3.1. Accuracy assessment

We evaluate the proposed method through the inter-participant experiment. For this purpose, the method is trained on 24 h of data from 32 participants. The testing phase consists of 24 h of data from 14 individuals. The OCSVM model is trained with "Reliable" samples and tested with both "Reliable" and "Unreliable" samples. To this end, we use 13338 "Reliable" 30-second PPG segments to train the model and 5329 "Reliable" and 14827 "Unreliable" segments to test it.

Fig. 9 shows the Receiver Operating Characteristic (ROC) curves and the Area Under Curve (AUC) values for the proposed and other ML-based methods. The ROC curve is plotted based on the model's true positive ratio (TPR) and false positive ratio (FPR). An ideal ROC curve has a TPR of 1 (top of the plot) and an FPR of zero (left of the plot). As indicated, the ROC curve of the proposed method bends significantly towards the upper left-hand corner compared to the baseline studies. Moreover, the AUC values of the proposed, supervised, unsupervised, and two deep learning methods are respectively 0.9971, 0.9929, 0.9841, 0.9875, and 0.9905. Based on the ROC curves and AUC values, our model performs better than the other methods in terms of FPR.

Table 6 presents the normalized confusion matrix for the proposed and baseline methods. We assume the "Reliable" class is the positive and the "Unreliable" class is the negative class. It is shown that the proposed method obtains the least false positive rate (0.01) and the most true negative rate (0.99). In contrast, Pereira et al. (supervised) and Shin et al. (deep learning) methods achieve the highest true positive rate (0.99). These two methods also have the lowest false negative rate (0.01).

The comparison of the methods in terms of precision, recall, F1-score, and accuracy is shown in Table 7. Precision shows the number of instances that are correctly classified among each class. As shown, the proposed method achieves the best precision for the "Reliable" class (0.99). In contrast, the supervised and two deep learning models have the highest precision for the "Unreliable" class (0.99). Recall indicates how many instances of each class have been classified correctly over the whole instances of that class. The proposed model's recall for the "Unreliable" class is 0.99, which is the highest value compared to the other methods. However, for the "Reliable" class, Pereira et al. (supervised) and Shin et al. (deep learning) methods obtain the best recall. F1-score is the harmonic mean between the precision and recall metrics. As indicated, our proposed method shows the best f1-score values for both "Reliable" (0.99) and "Unreliable" (0.98) classes. Accuracy is also the number of correctly classified instances out of all instances. The results show that the accuracy of the proposed method (0.97) is higher than other baseline methods.

In summary, our findings show that the proposed method performs best in detecting "Unreliable" PPG signals compared to the other techniques. In other words, due to the lowest false positive rate, the proposed method has the lowest false prediction of "Unreliable" signals. As a result, our approach ensures reliable data collection and analysis, enables precise vital signs extraction, and minimizes erroneous decision-making. In addition, due to accurate "Unreliable" signals detection, the system is able to discard "Unreliable" signals at the edge. This unreliable data removal optimizes data transmission (from the edge to the cloud) and minimizes edge device energy wastage.

6.3.2. Execution time assessment

We calculate the execution time of the methods on the two devices. The execution time indicates the device's processor and operating system performance. We compute the execution time of each method's parts (i.e., preprocessing, feature extraction, and classification) for 30-second PPG segments. The results are compared to determine how fast each device can perform each method.

The execution time (in milliseconds) of the proposed and baseline methods on Raspberry Pi 4 and Jetson Nano is indicated in Table 8. We calculate the average and standard errors of the execution time for 6712 PPG samples. The execution time to perform preprocessing, feature extraction, and classification, as well as the total time, have been presented for each method in each device. The fastest method is Vadrevu et al. (rule-based), with an execution time of 9.09 ms and 9.50 ms on Raspberry Pi 4 and Jetson Nano, respectively. In this method, the classification time indicates the execution time of the decision rules implementation. The proposed method is the second in terms of latency. The execution time is 24.71 ms and 26.35 ms on Raspberry Pi 4 and Jetson Nano, respectively. In contrast, Naeini et al. (deep learning) method has the highest latency among all methods, i.e., 249 ms on Raspberry Pi 4 and 202 ms on Jetson Nano. It should be noted that Pereira et al. (supervised) and Shin et al. (deep learning) techniques do not include any preprocessing step. Furthermore, no feature extraction is done in both deep learning approaches.

Table 6
Normalized confusion matrix of the proposed and baseline methods.

Method		Predicted: Reliable	Predicted: Unreliable
Vadrevu and Manikandan (2019) (Rule-based)	Actual: Reliable	0.70	0.30
	Actual: Unreliable	0.02	0.98
Mahmoudzadeh et al. (2021) (Unsupervised)	Actual: Reliable	0.94	0.06
	Actual: Unreliable	0.06	0.94
Pereira et al. (2019) (Supervised)	Actual: Reliable	**0.99**	**0.01**
	Actual: Unreliable	0.07	0.93
Naeini et al. (2019) (Deep learning)	Actual: Reliable	0.98	0.02
	Actual: Unreliable	0.07	0.93
Shin (2022) (Deep learning)	Actual: Reliable	**0.99**	**0.01**
	Actual: Unreliable	0.06	0.94
Proposed method	Actual: Reliable	0.88	0.12
	Actual: Unreliable	**0.01**	**0.99**

Table 7
Performance of the proposed and baseline methods.

Method	Class	Precision	Recall	F1-score	Accuracy	Support
Vadrevu and Manikandan (2019) (Rule-based)	Reliable	0.91	0.70	0.79	0.90	5329
	Unreliable	0.90	0.98	0.94		14 827
Mahmoudzadeh et al. (2021) (Unsupervised)	Reliable	0.86	0.94	0.90	0.94	5329
	Unreliable	0.98	0.94	0.96		14 827
Pereira et al. (2019) (Supervised)	Reliable	0.85	**0.99**	0.91	0.95	5329
	Unreliable	**0.99**	0.93	0.96		14 827
Naeini et al. (2019) (Deep learning)	Reliable	0.83	0.98	0.90	0.94	5329
	Unreliable	**0.99**	0.93	0.96		14 827
Shin (2022) (Deep learning)	Reliable	0.86	**0.99**	0.92	0.95	5329
	Unreliable	**0.99**	0.94	0.97		14 827
Proposed method	Reliable	**0.99**	0.90	**0.99**	**0.97**	5329
	Unreliable	0.96	**0.99**	**0.98**		14 827

Table 8
Raspberry Pi 4 and Jetson Nano execution time (ms) of the proposed and baseline methods.

Method	Device	Preprocessing	Feature extraction	Classification	Total
Vadrevu and Manikandan (2019) (Rule-based)	Raspberry Pi 4	2.22 (0.001)	6.90 (0.002)	0.03 (0.001)	**9.09** (0.004)
	Jetson Nano	2.15 (0.001)	7.33 (0.002)	0.02 (0.001)	**9.50** (0.004)
Mahmoudzadeh et al. (2021) (Unsupervised)	Raspberry Pi 4	2.20 (0.001)	122 (0.270)	0.91 (0.001)	125 (0.272)
	Jetson Nano	2.22 (0.001)	115 (0.241)	0.91 (0.001)	118 (0.243)
Pereira et al. (2019) (Supervised)	Raspberry Pi 4	0 (0)	169 (0.612)	1.68 (0.002)	170 (0.614)
	Jetson Nano	0 (0)	165 (0.583)	1.97 (0.002)	167 (0.585)
Naeini et al. (2019) (Deep learning)	Raspberry Pi 4	1.96 (0.001)	0 (0)	247 (0.287)	249 (0.288)
	Jetson Nano	1.95 (0.001)	0 (0)	200 (0.375)	202 (0.376)
Shin (2022) (Deep learning)	Raspberry Pi 4	0 (0)	0 (0)	194 (0.346)	194 (0.346)
	Jetson Nano	0 (0)	0 (0)	168 (0.316)	168 (0.316)
Proposed method	Raspberry Pi 4	1.70 (0.002)	21.50 (0.050)	1.53 (0.001)	**24.71** (0.053)
	Jetson Nano	1.72 (0.001)	23.10 (0.055)	1.54 (0.001)	**26.35** (0.057)

Average (Standard error).

Moreover, Fig. 10 illustrates the total execution time of Raspberry Pi 4 and Jetson Nano for each method. There is no significant difference between the two devices for proposed, rule-based, unsupervised, and supervised methods. However, the Jetson nano performs faster for the two deep learning approaches compared to Raspberry Pi 4.

Consequently, Vadrevu et al. (rule-based) was the best method in terms of execution time for the two devices, as this approach only includes a few simple decision rules. The proposed approach is the second fastest method on two devices. In contrast, the high complexity of deep learning models makes them the most time-consuming PPG SQA techniques on the devices.

6.3.3. Energy consumption assessment

We measure the energy consumption for the methods on the two devices. As previously mentioned, we use similar setups and environments for running the processes. We measure the voltage drop over the power line via a shunt resistor to calculate each

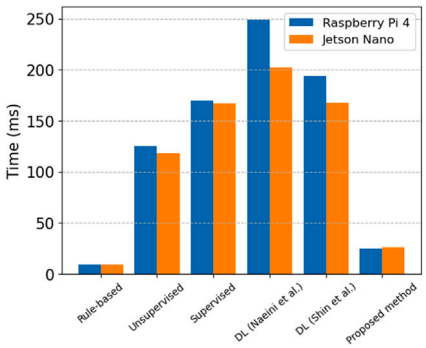
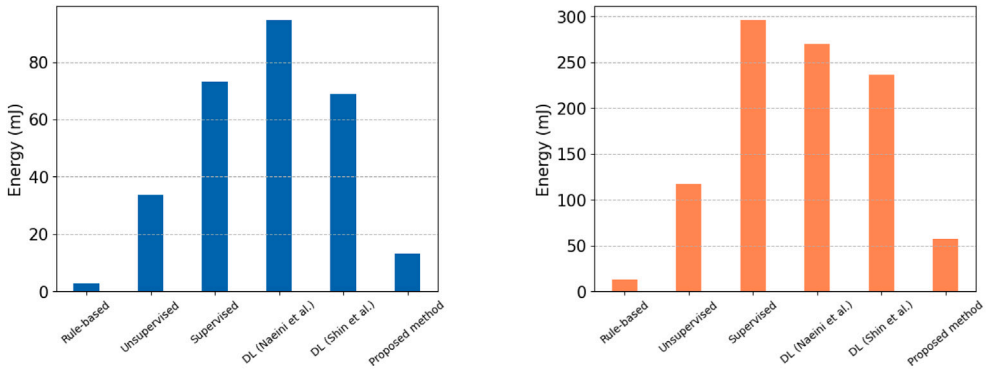


Fig. 10. Total execution time of the proposed and baseline methods on Raspberry Pi 4 and Jetson Nano.



(a) Raspberry Pi 4

(b) Jetson Nano

Fig. 11. Energy consumption on two embedded boards.

method's current and power consumption. Figs. 11(a) and 11(b) demonstrate the energy consumption of Raspberry Pi 4 and Jetson Nano, respectively. The measurements show that the Jetson Nano is generally a more power-consuming board compared to the Raspberry Pi 4. In idle mode, the Jetson Nano consumes an average of 3.10 W, while the Raspberry Pi 4 consumes 0.95 W. A similar power consumption ratio is also visible during the code execution. On average, the Jetson Nano consumes 3.9 times more power than Raspberry Pi 4. Since it is shown that the Jetson Nano performs faster than Raspberry Pi 4, we include the execution time in calculations, and instead of reporting the power consumption, we report the energy consumption of each method.

The developed code for each method consists of several functions that are called within some iterations, and each function has its unique profile for consuming the processing and memory resources. Therefore, as the results show, the overall power consumption of each method behaves uniquely. On both boards, Pereira et al. (supervised), Shin et al. (deep learning), and Naeini et al. (deep learning) methods are high-power-consuming methods. However, Vadrevu et al. (rule-based), the proposed, and Mahmoudzadeh et al. (unsupervised) methods consume much less power. The power consumption pattern of all methods for both boards is almost the same, except that on the Jetson Nano, Pereira et al. (supervised) method is slightly more power-consuming than the Naeini et al. (deep learning) method compared to Raspberry Pi 4.

The measurements show that the proposed method uses the boards' resources intensively in a very short time, so the power consumption of the proposed method is higher than other methods. However, its fast execution time compensates for the energy consumption. It results in a low energy consumption profile, which is the second-best energy-efficient method on both Raspberry Pi 4 and Jetson Nano.

The proposed method outperforms the other methods overall. It obtains the best accuracy and true negative rate (specificity). It also has the lowest execution time and energy consumption compared to the other ML-based methods. The Pereira et al. (Supervised) and Shin et al. (Deep learning) methods have the highest true positive rate (sensitivity). However, these methods are highly time- and energy-consuming. On the other hand, the Vadrevu et al. (rule-based) method is the fastest and least energy-consuming method (slightly better than the proposed method). However, it has the lowest accuracy compared to the other methods. Consequently, the findings show that our approach provides an acceptable trade-off between accuracy, time, and energy. Our proposed method shows significant promise for real-time vital signs monitoring in resource-limited environments, due to its ability to achieve high levels of accuracy while utilizing minimal resources.

7. Conclusion

Noise is a prevalent problem in PPG data collection. It distorts the signals and, subsequently, causes failure in decision-making. In addition, unreliable data collection wastes a large amount of energy in wearable devices. In this paper, we proposed a lightweight PPG SQA method to detect "Reliable" and "Unreliable" PPG signals on battery-powered embedded systems. We first investigated forty-five features extracted from PPG signals and selected the best five features in terms of accuracy and execution time. Then, we developed a one-class SVM to classify "Reliable" and "Unreliable" signals. The proposed method was assessed using a PPG dataset collected via smartwatches from 46 individuals in free-living conditions. We compared our method with five existing methods published recently. The methods were evaluated in terms of accuracy, execution time, and energy consumption on two embedded devices. The proposed method outperformed the other methods overall. It obtained the best accuracy and true negative rate (specificity). It also had the lowest execution time and energy consumption compared to the other ML-based methods. The results indicated that the proposed method provided an acceptable and good trade-off between accuracy, time, and energy. As future work, we will assess the efficacy of our proposed PPG SQA method on wearable devices, such as smartwatches and smart rings. We also intend to evaluate the proposed method using a PPG dataset collected in an extended period spanning several weeks. In addition, our findings are limited to individuals who are in good health since we only considered data from healthy participants in our study. Cardiovascular conditions (e.g., atrial fibrillation) affect the PPG signal, which may impact the performance of the proposed SQA method. To address this limitation, our future work in this direction will consider PPG data collected from individuals with different health conditions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This research received support from the Academy of Finland through the SLIM Project (grant numbers 316810 and 316811) and the U.S. National Science Foundation (NSF) through the UNITE Project (grant number SCC CNS-1831918).

References

Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Azar, Joseph, Makhoul, Abdallah, Couturier, Raphael, & Demerjian, Jacques (2021). Deep recurrent neural network-based autoencoder for photoplethysmogram artifacts filtering. *Computers & Electrical Engineering*, *92*, Article 107065.

Boyd, Stephen, & Vandenberghe, Lieven (2004). *Convex optimization*. Cambridge University Press, Google Scholar Google Scholar Digital Library Digital Library.

Castaneda, Denisse, Esparza, Aibhlin, Ghamari, Mohammad, Soltanpur, Cinna, & Nazeran, Homer (2018). A review on wearable photoplethysmography sensors and their potential future applications in health care. *International Journal of Biosensors & Bioelectronics*, *4*(4), 195.

Chong, Jo Woon, Dao, Duy K., Salehizadeh, S. M. A., McManus, David D., Darling, Chad E., Chon, Ki H., et al. (2014). Photoplethysmograph signal reconstruction based on a novel hybrid motion artifact detection–reduction approach. Part I: Motion and noise artifact detection. *Annals of Biomedical Engineering*, *42*(11), 2238–2250.

Deepu, C., et al. (2016). A hybrid data compression scheme for power reduction in wireless sensors for IoT. *IEEE Transactions on Biomedical Circuits and Systems*, *11*(2), 245–254.

Domingues, Rémi, Filippone, Maurizio, Michiardi, Pietro, & Zouaoui, Jihane (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, *74*, 406–421.

Elgendi, Mohamed, Norton, Ian, Brearley, Matt, Abbott, Derek, & Schuurmans, Dale (2013). Systolic peak detection in acceleration photoplethysmograms measured from emergency responders in tropical conditions. *PLoS One*, *8*(10), Article e76585.

Gay, Warren (2014). *Raspberry Pi hardware reference*. A Press.

Guyon, Isabelle, Weston, Jason, Barnhill, Stephen, & Vapnik, Vladimir (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1), 389–422.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Jolliffe, Ian T., & Cadima, Jorge (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society of London. Series A. Mathematical, Physical and Engineering Sciences*, *374*(2065), Article 20150202.

King, Christine E., & Sarrafzadeh, Majid (2018). A survey of smartwatches in remote health monitoring. *Journal of Healthcare Informatics Research*, *2*(1), 1–24.

Kyriacou, Panicos A., & Allen, John (2021). *Photoplethysmography: technology, signal analysis and applications*. Academic Press.

Li, Qiao, & Clifford, Gari D. (2012). Dynamic time warping and machine learning for signal quality assessment of pulsatile signals. *Physiological Measurement*, *33*(9), 1491.

Li, K., et al. (2011). Onboard tagging for real-time quality assessment of photoplethysmograms acquired by a wireless reflectance pulse oximeter. *IEEE Transactions on Biomedical Circuits and Systems*, *6*(1), 54–63.

Liu, Shing-Hong, Li, Ren-Xuan, Wang, Jia-Jung, Chen, Wenxi, & Su, Chun-Hung (2020). Classification of photoplethysmographic signal quality with deep convolution neural networks for accurate measurement of cardiac stroke volume. *Applied Sciences*, *10*(13), 4612.

Liu, S., et al. (2010). Heart rate extraction from photoplethysmogram on fuzzy logic discriminator. *Engineering Applications of Artificial Intelligence*, *23*(6), 968–977.

Mahmoudzadeh, Aysan, Azimi, Iman, Rahmani, Amir M, & Liljeberg, Pasi (2021). Lightweight photoplethysmography quality assessment for real-time IoT-based health monitoring using unsupervised anomaly detection. *Procedia Computer Science*, *184*, 140–147.

Makowski, Dominique, Pham, Tam, Lau, Zen J., Brammer, Jan C., Lespinasse, François, Pham, Hung, et al. (2021). NeuroKit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, *53*(4), 1689–1696. http://dx.doi.org/10.3758/s13428-020-01516-y.

Mehrabadi, M., et al. (2020). Sleep tracking of a commercially available smart ring and smartwatch against medical-grade actigraphy in everyday settings: instrument validation study. *JMIR mHealth and uHealth*, *8*(11), Article e20465.

Merkel, Dirk (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, *2014*(239), 2.

Moraes, Jermana L., Rocha, Matheus X., Vasconcelos, Glauber G., Vasconcelos Filho, José E., De Albuquerque, Victor Hugo C., & Alexandria, Auzuir R. (2018). Advances in photopletysmography signal analysis for biomedical applications. *Sensors*, *18*(6), 1894.

Naeini, Emad Kasaeyan, Azimi, Iman, Rahmani, Amir M, Liljeberg, Pasi, & Dutt, Nikil (2019). A real-time PPG quality assessment approach for healthcare internet-of-things. *Procedia Computer Science*, *151*, 551–558.

Orphanidou, Christina (2017). Signal quality assessment in physiological monitoring: state of the art and practical considerations.

Pereira, Tania, Gadhoumi, Kais, Ma, Mitchell, Liu, Xiuyun, Xiao, Ran, Colorado, Rene A, et al. (2019). A supervised approach to robust photoplethysmography quality assessment. *IEEE Journal of Biomedical and Health Informatics*, *24*(3), 649–657.

Plawrence (2019). *Jetson nano developer kit*. NVIDIA.

Reddy, G., et al. (2020). On-device integrated ppg quality assessment and sensor disconnection/saturation detection system for IoT health monitoring. *IEEE Transactions on Instrumentation and Measurement*, *69*(9), 6351–6361.

Reddy, G., et al. (2022). Evaluation of objective distortion measures for automatic quality assessment of processed PPG signals for real-time health monitoring devices. *IEEE Access*, *10*, 15707–15745.

Roy, M., et al. (2020). Photoplethysmogram signal quality evaluation by unsupervised learning approach. In *2020 IEEE applied signal processing conference* (pp. 6–10). IEEE.

Samsung gear sport watch. https://www.samsung.com/us/mobile/wearables/smartwatches/gear-sport-blue-sm-r600nzbaxar/.

Schölkopf, Bernhard, et al. (1999). Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, *12*.

Shin, Hangsik (2022). Deep convolutional neural network-based signal quality assessment for photoplethysmogram. *Computers in Biology and Medicine*, *145*, Article 105430.

Simonyan, Karen, & Zisserman, Andrew (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Vadrevu, Simhadri, & Manikandan, M. Sabarimalai (2019). Real-time PPG signal quality assessment system for improving battery life and false alarms. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *66*(11), 1910–1914.

Van Engelen, Jesper E., & Hoos, Holger H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440.

Villa-Pérez, Miryam Elizabeth, Alvarez-Carmona, Miguel A., Loyola-González, Octavio, Medina-Pérez, Miguel Angel, Velazco-Rossell, Juan Carlos, & Choo, Kim-Kwang Raymond (2021). Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. *Knowledge-Based Systems*, *218*, Article 106878.

Virtanen, Pauli, Gommers, Ralf, Oliphant, Travis E., Haberland, Matt, Reddy, Tyler, Cournapeau, David, et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Welch, Peter (1967). The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, *15*(2), 70–73.