Data-driven Energy-efficient Adaptive Sampling Using Deep Reinforcement Learning

BERKEN UTKU DEMIREL, LUKE CHEN, and MOHAMMAD ABDULLAH AL FARUQUE,
University of California Irvine, USA

This article presents a resource-efficient adaptive sampling methodology for classifying electrocardiogram (ECG) signals into different heart rhythms. We present our methodology in two folds: (*i*) the design of a novel real-time adaptive neural network architecture capable of classifying ECG signals with different sampling rates and (*ii*) a runtime implementation of sampling rate control using deep reinforcement learning (DRL). By using essential morphological details contained in the heartbeat waveform, the DRL agent can control the sampling rate and effectively reduce energy consumption at runtime. To evaluate our adaptive classifier, we use the MIT-BIH database and the recommendation of the AAMI to train the classifiers. The classifier is designed to recognize three major types of arrhythmias, which are supraventricular ectopic beats (SVEB), ventricular ectopic beats (VEB), and normal beats (N). The performance of the arrhythmia classification reaches an accuracy of 97.2% for SVEB and 97.6% for VEB beats. Moreover, the designed system is 7.3× more energy-efficient compared to the baseline architecture, where the adaptive sampling rate is not utilized. The proposed methodology can provide reliable and accurate real-time ECG signal analysis with performances comparable to state-of-the-art methods. Given its time-efficient, low-complexity, and low-memory-usage characteristics, the proposed methodology is also suitable for practical ECG applications, in our case for arrhythmia classification, using resource-constrained devices, especially wearable healthcare devices and implanted medical devices.

CCS Concepts: • **Computer systems organization** → *Embedded software*; • **Mathematics of computing** → *Dimensionality reduction;*

Additional Key Words and Phrases: Adaptive sampling, energy-efficient, heart rate, real-time, wearable devices

ACM Reference format:
Berken Utku Demirel, Luke Chen, and Mohammad Abdullah Al Faruque. 2023. Data-driven Energy-efficient Adaptive Sampling Using Deep Reinforcement Learning. *ACM Trans. Comput. Healthcare* 4, 3, Article 19 (September 2023), 19 pages.
https://doi.org/10.1145/3598301

19

1 INTRODUCTION

Cardiovascular disease (CVD) is one of the leading causes of mortality worldwide. According to a recent work [2], the prevalence of CVDs is 48% for adults in the United States. In addition, according to the National Center for Health Statistics, the age-adjusted death rate of CVDs was 219.4 per 100,000 in 2017, which equalled 859,125 dead and 2.2 million people hospitalized in the United States [43]. It is now well established from various studies that the death toll due to CVDs is more than cancer and chronic lung disease combined [3]. Furthermore, productivity losses and medical costs of CVD were 555 billion US dollars in 2015. It is expected to reach $1.1 trillion in 2035

B. U. Demirel and L. Chen contributed equally to this research.

Authors' addresses: B. U. Demirel, L. Chen, and M. A. Al Faruque, University of California Irvine, Irvine, CA; emails: bdemirel.com, pangwang@uci.edu, alfaruqu@uci.edu.

2637-8051/2023/09-ART19 $15.00
https://doi.org/10.1145/3598301

[10]. Although the CVDs include arrhythmia, heart defect, dilated cardiomyopathy, and valvular heart disease, arrhythmias are the most important ones among them, as they result in heart failure, heart attack, and sudden cardiac arrest. The higher risk associated with recurrent heart attacks requires continuous monitoring of those patients. Moreover, 1 out of 5 heart attacks is silent, where victims are unaware of the damage [1]. The mortality rate significantly increases by 41%–62% if the treatment is delayed for more than two hours from MI initiation [18]. The facts mentioned above show the importance of real-time monitoring and detecting of arrhythmias for patients. Currently, most of the monitoring occurs in a clinical environment with bulky medical equipment that lacks portability. Therefore, wearable devices represent a more convenient solution for continuous monitoring daily [32, 36]. Hence, the need for developing long-lasting energy-efficient devices for cardiac activity monitoring or classification is compelling.

The use of data-driven techniques in healthcare applications is popular given the success of deep learning algorithms and their ability to learn and adapt to the complex and dynamic environments present in healthcare applications [11, 27]. Decision-making and control optimization problems such as dynamic treatment, automated diagnosis, and so on, are prominent in the automated healthcare systems. Traditional control-theory and supervised learning methods have been applied to tackle these problems, however, the former requires sophisticated mathematical models that are often unavailable due to the complexity and partial information of the environment, while the latter can learn given enough data, but require expert labels that may not always be available. Deep Reinforcement Learning differs from them in that it does not require mathematical modeling of the environment, nor does it need expert labels to learn an optimal control policy. DRL constructs adaptive controllers capable of making optimal decisions by utilizing the contextual information in healthcare systems [49]. DRL solves sequential-decision problems by taking into account the state observations of the system and learns through trial-and-error the optimal action to take that will maximize the long-term expected reward based on the current context and past experiences. State information in the form of the statistical, time-domain, frequency domain, and other physiological features, like the morphology of ECG, can provide useful contextual information to the learning agent.

Electrocardiogram (ECG) signal, which records the electrical signals of heart activity, is one of the most effective and available methods for detecting cardiac arrhythmias [40], since the ECG measurement is a non-invasive, simple, economical, fast, and safe method that can be accessed in hospitals, healthcare devices, and wearable devices. Although wearable intelligent devices for long-term ECG monitoring have recently been proposed, the main limitations of these algorithms are their memory and energy consumption burden on the devices, limiting the continuous monitoring of patients. Recent developments in cloud computing paved the way to analyze long-term recorded ECG signals offline using remote cloud servers. The proposed algorithms in the cloud provide powerful classification performance. However, they cannot be implemented on resource-constrained devices due to their memory requirements and high energy consumption [24, 35]. Moreover, since all computing occurs in the cloud, the latency of the system increases, which endangers the users' cardiac situation, as it becomes an offline classification algorithm rather than a real-time one. Furthermore, the raw ECG data recorded from wearable devices is transferred to a mobile phone or cloud system for further investigation. This process of offloading raw data from wearable devices to mobile phones consumes tremendous energy for communication, and as a result, reduces the battery life of wearable devices, as well as mobile phones [5, 26], which ultimately decreases the monitoring time of patients.

To address the above-mentioned challenges, this article proposes a novel method for adaptive sampling of ECG signals in runtime using deep reinforcement learning on a resource-constrained device to decrease energy consumption of transmission and classification, thereby increasing the monitoring time of patients.

1.1 Motivational Example

We have conducted several experiments to show why adaptive sampling is needed and can be helpful in classifying ECG signals for resource-constrained devices. We have created five different classifiers, one for each

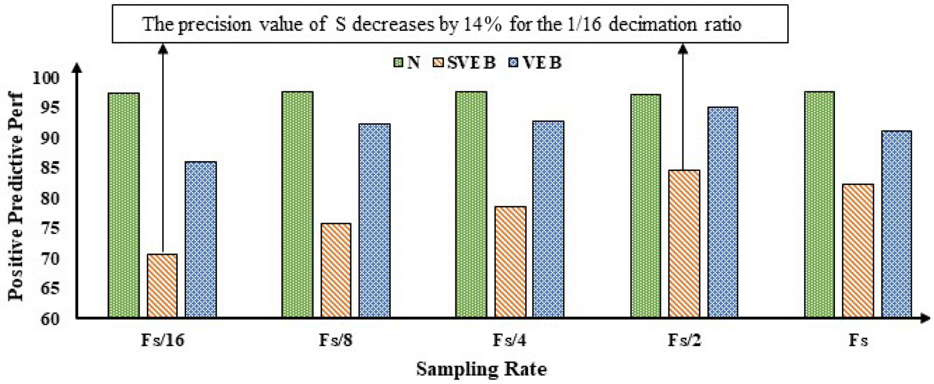

Fig. 1. The classification performance of each beat type under different sampling rates.

decimation ratio of the original sampling rate ($Fs/16$, $Fs/8$, $Fs/4$, $Fs/2$) and one with the actual sampling rate of the dataset (Fs), to detect N, SVEB (S), and VEB (V) in MIT-BIH dataset [29]. While performing the experiments, we followed the common evaluation method, which is the intra-patient paradigm [22, 51], in the literature. For each decimation rate and beat class, the positive predictive value ($TP/(TP + FP)$) in percentage is obtained and reported in Figure 1.

The most interesting aspect of Figure 1 is that the classification performance of the neural network stays relatively constant, around 97.6%–97.7%, for **N-type (Non-ectopic)** beats in each sampling rate. However, S-type (**Supraventricular Ectopic Beats (SVEB)**) classification rate decreases heavily in low sampling rates, especially $Fs/16$ with a 14% drop. Although it looks like V-type (**Ventricular Ectopic Beats (VEB)**) detection performance stays close to constant in different sampling rates, its percentage is decreased by 9% from the best sampling rate for the V-type. Surprisingly, when we decrease the sampling rate from Fs to Fs/2, resulting in a resolution decrease for beats, the classifier performance increases for VEB and SVEB beats while not changing for N types. Although at first glance this result is unexpected, our findings reflect those of Zhai et al. [51], who also found that the positive predictive value of S beats is sensitive to the input size and can increase with a smaller input size. Although decreasing the sampling rate in ECG beats leads to information loss, it is quite common to increase the performance by losing redundant information in ECG. For example, a recent paper showed [23] that ignoring beats from different leads can improve the performance of the classifiers. Also, it should be noted that the classifier designed with $Fs/8$ has 8× fewer parameters and **Floating point operations (FLOPs)** compared to the baseline model that is used for the original sampling rate. Overall, these results suggest that if the sampling rate of the ECG signals is controlled precisely during runtime, then significant energy consumption can be prevented without sacrificing the classification performance. Even, as can be seen from Figure 1, the classification performance can be increased for VEB and SVEB if the sampling rate is chosen adaptively during inference instead of a constant Fs.

1.2 Research Challenges

The idea of using an adaptive sampling rate with a **CNN (Convolutional Neural Network)** architecture utilizing the Reinforcement Learning during inference phase instead of the constant input size introduces the following research challenges:

- The input states of the reinforcement learning model should provide sufficient context needed for deciding the sampling rate at runtime. The extraction of the required information from raw data should be lightweight so its associated resource overhead does not overshadow the gains achieved by the adaptive sampling.

- The RL model execution time should be negligible as to enable real-time classification performance.
- The adaptive system should be energy-efficient while maintaining comparable classification performance to the baseline architectures in state-of-the-art works.

1.3 Novel Contributions

The novel contributions of this article are as follows:

- An arrhythmia detection method using a sampling rate-adaptive CNN architecture that reaches or outperforms the state-of-the-art works on the well-known MIT-BIH datasets [29] in resource-constrained devices while being energy- and memory-efficient.
- Development of reinforcement learning environment that discerns the contextual and temporal correlations existing in the consecutive ECG segments to determine the rate of decimation to be applied to the digitized signal for decreasing energy-consumption and latency while achieving comparable classification performance. To the best of our knowledge, no state-of-the-art work has considered the current signals' physiology to design such an adaptive system.
- Evaluation in terms of computational efficiency shows that our solution is 7.3× more energy-efficient compared to our baseline where the sampling rate is constant. Moreover, evaluation on the hardware demonstrates that our proposed solution is compatible with low-memory devices with a minimum 128 KB of RAM.
- All of the analysis and results presented in this article are openly available at https://github.com/Berkendemirel/AdaptiveSampling-DRL for further research and experimentation. Moreover, we have included the RL agent with weights that can easily be run on another machine and potentially reused on a new dataset.

The rest of the article is organized as follows: In Section 2, we review related works for changing sampling rate in heart monitoring systems adaptively. Section 3 describes the proposed method. Section 4 shows the experimental setup. Section 5 discusses the experimental results and analysis. Finally, a conclusion is drawn in Section 8.

2 RELATED WORKS

A large volume of published studies investigates the role of sampling rate in finding the fiducial points (R peak or QRS complex) in the ECG waveform or classifying ECG signals into different cardiac diseases. For instance, to determine the effects of input size or resolution on the performance, Zhai et al. [51] compared six different input sizes of CNN classifiers for detecting arrhythmic beats. They found that the classification performance for S-type is the lowest degree at a small input size, which is likely due to the low resolution of input to capture the necessary information of the original ECG signal. Yet, detection performance for V-type beat remains relatively high, probably because V-type beats are usually well distinguished from other beat types. The adaptive sampling can be seen as a non-uniform feature extraction/selection in the time domain for the Deep Learning models, since the sampled signals are fed to the CNN or LSTM Long Short Term Memory for the classification [51]. The non-uniform feature extraction was investigated before in the frequency domain instead of the time domain. For example, authors in Reference [6] proposed to analyze the ECG signal only in the frequency domain by acquiring more features in the low-frequency spectrum and fewer features in the high-frequency. After selecting features in the frequency domain, the authors proposed a fast spectral artificial neural network to classify ECG signals.

Also, as the sampling rate of the signal directly affects the memory requirement and the energy consumption of a device, much of the research for energy-efficient algorithms has focused on controlling the sampling frequency during runtime. For example, a recent study from Demirel et al. [7] has proposed to use a sampling rate regulator that is just controlled by binary classifier output, for changing the sampling rate of ECG signals to the degree that the classification performance stays constant while decreasing the energy consumption of a classifier that

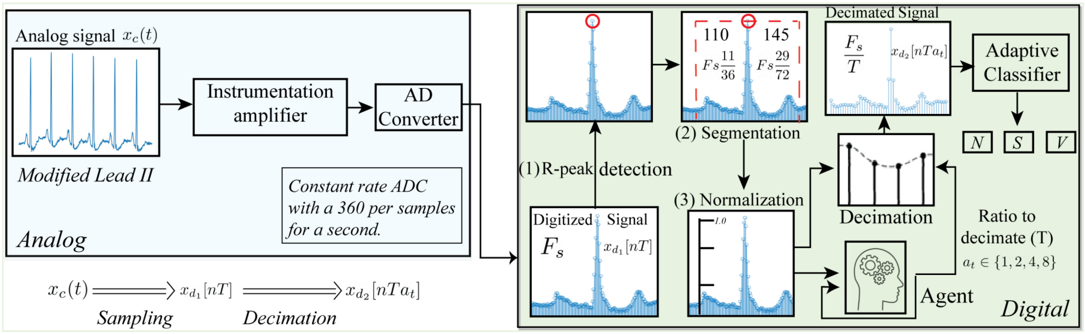

Fig. 2. Overview of adaptive sampling rate using DRL.

runs on a resource-constrained device. However, the action space of this controller is minimal, as the overall system is designed only for two different sampling rates (high or low). Moreover, since the change in sampling rate is only controlled by classifying the current heartbeat as normal or abnormal, the controller does not utilize the information about the present and past waveforms or heart rate variability to change the sampling rate of the current waveform in a more realistic scheme. Since the transmission of the raw ECG signals from sensors to mobile phones or cloud servers introduces additional energy consumption to the system, there is a large volume of published studies describing the role of **compressive sensing (CS)** [39] to decrease the overall energy consumption. CS is a widely used technique that joins both sampling and compression to decrease the intensity of information obtaining and transmission [9]. The main difference between adaptive sampling and CS is that compressed signals need to be reconstructed or sparse coded for further usage. While the proposed adaptive sampling method in this work requires no post-processing of the signals.

In another study by Augustyniak [4], the authors proposed to sample the ECG signal adaptively by allocating space in the output data stream accordingly to the information density in the input series by using an encoder and decoder. Eventually, the authors showed that the proposed adaptive sampling, which has two parts: detection of heartbeats (QRS complex) and detection of ECG wave borders, preserves the essential sections of the heartbeat and maintains all diagnostic features of the original signal for reconstructing the initial ECG. Although these two works focused on reconstructing the original sampled signal, recent outcomes show that using a sampling frequency below the theoretically required Nyquist rate can be used to classify cardiovascular diseases. For example, in contrast to these studies, a recent work conducted by Zanoli et al. [50] presented that the QRS complex of ECG signals can be detected using a sampling frequency that is much below the theoretically required Nyquist rate. By this technique, which is called event-based sampling, where the events represent the beats of the heart, they have drastically reduced the average sampling frequency of the signal and, hence, the energy needed to process it and extract the relevant heart rate information from ECG signals. Although the heart rate variability features are helpful for the detection of abnormalities such as arrhythmias in ECG signals, its usage is limited for a wide range of diseases. And, since they only aimed to detect the QRS complex in the complete signal, the morphological waveform features are lost due to low sampling frequency. Therefore, developing a lightweight runtime adaptive sampling controller is needed for realizing low-power continuous ECG monitoring devices.

3 PROPOSED METHODOLOGY

The following sections provide the details of our proposed method. The overview of our proposed method is demonstrated in Figure 2. The proposed method can be divided into two main sections. The first one is the analog section where the continuous ECG signals ($x_c(t)$) are amplified and converted into a digital domain using a constant rate ADC with a 360 Hz. The second Digital section starts with pre-processing of the digitized signal ($x_{d_1}[nT]$) with R-peak detection. Then, it continues with segmentation and normalization of heartbeats to

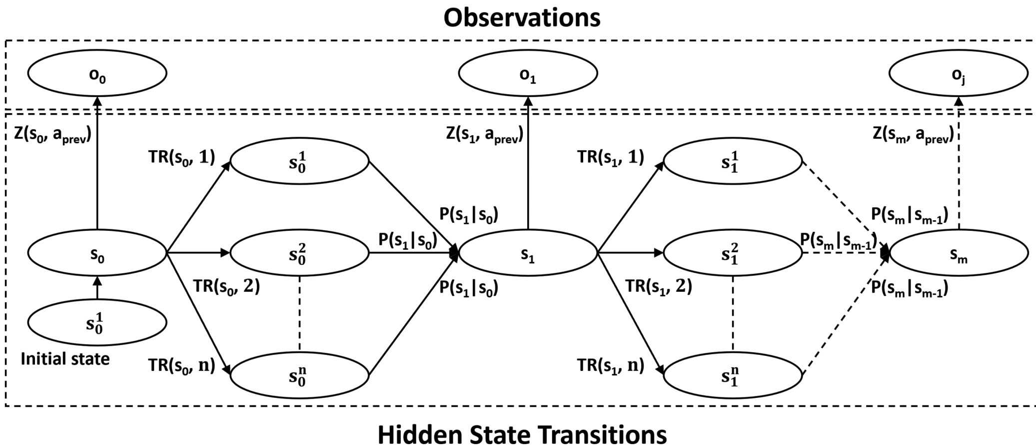

Fig. 3. POMDP state diagram.

decide the decimation ratio of the current beat. Finally, the decimated signal ($x_{d_2}[nTa_t]$) is fed to the adaptive classifier for beat detection. Finally, the adaptive classifier gives one of the three classes, which are Non-ectopic (N), Supraventricular Ectopic Beats (S), and Ventricular Ectopic Beats (V) as output for each beat. The detailed description of each part is given in the following sections.

3.1 Pre-processing Steps

3.1.1 ECG Lead Selection. This study only used the modified lead II channel from the MIT-BIH database [29], similar to other works [22, 51]. Automatic cardiac activity classification based on ECG is beneficial for portable or wearable devices, and it is known that few channel numbers (even single-channel) would be found in these devices. Hence, we developed our algorithm to run with a single channel of ECGs.

3.1.2 R-peak Detection. Although numerous robust methods have already been available for R peak detection [31, 33], the R peaks have been labeled for the corresponding sample in the MIT-BIH dataset. Therefore, we used the annotated R peak locations from the dataset without applying any peak detection algorithms. The algorithm for improving R-peak detection is beyond the scope of this manuscript.

3.1.3 Segmentation. Similar to other work [30], we take 110 points before the peak and 145 points after the peak (totally 256 points containing the R peak), which are used to represent the corresponding heartbeat.

3.1.4 Normalization. As the last step for preprocessing, the segmented beats are normalized to have a maximum value of 1 before feeding them to the RL agent and adaptive classifier.

3.2 Reinforcement Learning (RL)

We formulate the adaptive sampling rate problem as a **partially observable Markov decision process (POMDP)** defined by the tuple $<S, A, O, TR, Z, R>$. Here, $S \subseteq \mathbb{R}^m$ is the set of states, $A \subseteq \mathbb{R}^n$ is the set of actions, $O \subseteq \mathbb{R}^j$ is the set of observations, $TR : S \times A \rightarrow S$ is the transition function, $Z : S \times A \rightarrow O$ is the observation function, and $R : O \times A \rightarrow \mathbb{R}^k$ is the immediate reward function. A learned policy $\pi : O \rightarrow A$ maps observations to actions over a time horizon $T \in \mathbb{N}$, which is used to generate an action trajectory $(a_0, a_1, \ldots, a_T)$. Details on the POMDP state transitions can be observed in Figure 3, where $s_0, s_1, \ldots, s_m$ represents the underlying ECG signal states and $s_0^1, s_0^2, \ldots, s_0^n$ represents the corresponding decimated signal states. $P(s'|s)$ describes the cardiac/beat to beat transition probability of the underlying ECG time-series data, and a_{prev} denotes the previously chosen action.

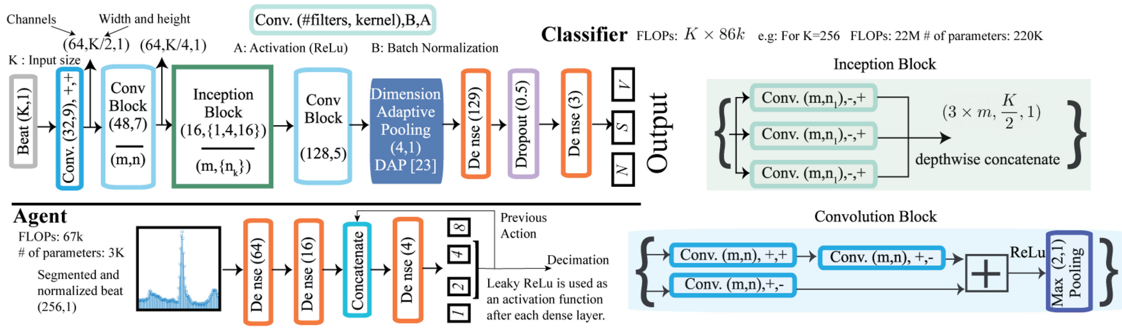

Fig. 4. Overview of proposed classifier (above) with reinforcement learning network (below). The "+" and "−" signs near the convolutional layer show whether the batch normalization and/or activation are applied.

We consider a standard **Reinforcement Learning (RL)** setup leveraging the **Double Q-Learning (DQL)** [42] algorithm that operates in an environment E. At each discrete timestep t, the agent observes $o_t = Z(s_t, a_{t-1})$, picks an action $a_t = \pi(o_t)$, and receives a scalar reward $r_t = R(s_t, a_t)$ and the next observation $o_{t+1} = Z(s_{t+1}, a_t)$ from E.

The objective of the DQL agent is to learn the estimates of optimal action values, or Q-values, through iterative updates of a DNN, known as the policy network, to approximate the Q function that maps any s to a vector of action values. The policy network is updated using Equation (1),

$$Q(s_t, a_t) = r_t + \gamma Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta_t); \theta'_t), \tag{1}$$

which comprises the immediate reward r_t, received by taking a_t given s_t, and the expected total discounted future rewards, represented by the product of γ, a discount factor $\in [0, 1]$, and the Q-value of s_{t+1}. A second Q function known as the target network, represented by the network weights θ'_t is used according to Reference [42] to reduce Q-value overestimation.

Agent. We use a single RL agent to control the decimation ratio during runtime. During exploration, the agent aims to maximize the overall reward of one episode that we define as one subject session, where each session duration varies according to the number of collected data samples per subject. As shown in Figure 4, the DQL agent comprises two hidden layers of size 128 and 16 + 1, with a four-node output-layer representing the value estimate of each action for any s such that the optimal action is $a = \arg\max_a Q_\pi(s, a)$ for the learned policy π. During our experiments, we investigated several architectures for DQL agents through grid search over the number of layers (1–4) and hidden nodes (2–512) such that the total memory usage does not exceed the runtime constraints of the embedded device. It is noted that minimal to no policy improvements have been observed when we increase the number of layers and nodes for the agent. Therefore, the architecture with two hidden layers is used to make it efficient in terms of energy and execution latency. We concatenate the observation of the previous action with the current ECG feature embeddings in the last hidden layer following a similar technique in Reference [48] where it was shown that the concatenation of intermediate features with past observations improved the performance compared to standard DQL. Instead of directly concatenating a low-dimension feature (the previous action of size 1) with a high-dimension feature (an ECG segment of size 256), applying smaller intermediate-layers acts as an encoder or a pooling layer of a CNN whereby higher feature abstractions can be extracted while also achieving dimension reduction, which helps balance the influence of high-dimension features and low-dimension features during the learning process.

State Observation. At every timestep, the agent observes a segmented beat of size 256 that is pre-processed according to Section 3.1. Alongside the ECG segment, we also include the previous action into the observation and take advantage of the inherent temporal correlation between consecutive ECG segments in the decision process. It is apparent that consecutive ECG segments are similar such that the action applied on a correctly classified previous beat can likely be applied to the current beat with correct classification.

Action. The agent generates four different actions to control the sampling rate during runtime. The discrete action space is chosen rather than continuous, because it is observed that the small changes in the sampling rate have no notable effects on both energy and accuracy and make the environment unnecessarily complex, which results in inefficient exploration during training. The actions for decimation, which is the extraction of every Nth sample from the digitized signal, are chosen as $a \in \{1, 2, 4, 8\}$. In other words, if the agent chooses the action 8 during runtime, then the digitized signal is decimated by 8 before sending it to the classifier. The overall process is explained in Equation (2):

$$x_c(t) \xrightarrow{\text{Sampling}} x_{d_1}[nT] \xrightarrow{\text{Decimation}} x_{d_2}[nTa_t], \tag{2}$$

where the x_c, x_{d_1}, x_{d_2} are the continuous, digitized, and decimated signals, respectively. And, the T represents the original sampling rate of the signal from the **Analog-to-Digital (ADC)** converter.

Reward. The reward function of the agent considers both accuracy and energy savings, which are both affected by the decision made by the agent. The accuracy portion aims to maximize the average classification of all events (heartbeats) in the environment (patients). The energy-efficiency goal of the agent is to keep the overall architecture FLOPs as minimum as possible. As the number of FLOPs are decreasing when the decimation ratio is increased, the reward function is scaled by the inverse of the decimation. Considering these two objectives, the reward for the agent is defined as follows:

$$R_{agent} = \begin{cases} \lambda R_{FLOPs} & \text{if the classification is correct} \\ -p & \text{otherwise,} \end{cases} \tag{3}$$

where λ is the reward scaling factors and p is the penalty value, when the classification is correct, the reward is scaled with FLOPs improvement. Otherwise, the reward is a negative value to punish the agents. The λ and p values were found empirically and set to unit value and 10, respectively. We normalize the reward between [−1, 1] to improve the convergence of the RL learning process. The R_{FLOPs} is defined as a decrease in the number of FLOPs for each action where the FLOPs are calculated as mentioned below.

FLOPs Computation. While calculating the total number of **floating-point operations (FLOPs)**, we have followed Reference [28], where the convolution is assumed to be implemented as a sliding window and that the nonlinearity function is computed for free. For convolutional kernels, we have:

$$\text{FLOPs} = 2HW(C_{in}K + 1)C_{out}, \tag{4}$$

where H, W, and C_{in} are the height, width, and number of channels of the input feature map, K is the kernel width, and C_{out} is the number of output channels. As we have only a single channel, the height H of the convolutional kernel is taken as 1. For fully connected layers, we compute FLOPs as:

$$\text{FLOPs} = (2I - 1)O, \tag{5}$$

where I and O are the input and output dimensionality, respectively. As the output dimension of the adaptive pooling layer is the same regardless of decimation rate, the FLOPs for the last two dense layers are exempted while calculating the reduction by the agent's decision. When the agent chooses the decimation rate among the $\{1, 2, 4, 8\}$ values in the action space, the W of the input is decreased by the respective action value. For example, if the agent chooses the action 8, then the size of the input beat decreases to $K/8$ from K, resulting in an 8× reduction in FLOPs (i.e., $R_{FLOPs} = 8$).

3.3 Adaptive CNN Architecture

The designed classifier is a convolutional neural network, which takes as input only the raw heartbeat samples and no other patient- or ECG-related features and classifies a single heartbeat into three classes (N, SVEB, VEB). The architecture is designed to extract various morphological features from the complete beat, such as P and T waveforms, by employing different lengths of convolutional filters at different layers of the architecture. While designing the architecture, we utilized both the Residual connections introduced by He et al. in Reference [16] and the Inception architecture [41] that has been shown to achieve good performance while maintaining computational and memory costs at low levels.

The Conv blocks in Figure 4 show the implemented original residual connections where the activation is applied after addition. The model includes two residual blocks with different kernel sizes and filter numbers. Every residual block subsamples its inputs by a factor of 2 by taking the maximum sample (i.e., max pooling with stride 2). The **Rectified Linear Unit (ReLu)** is utilized as the activation function in the classifier. All convolutional layers are implemented using a stride of 1 except the first filter, which moves two samples, resulting in half of the samples $K/2$. Unlike the Inception architecture, the wider layer (one after the first residual block) is not stacked up together; instead, we have used Residual blocks, which helps to reduce the dimension of the network while combining the various features of a heartbeat. Otherwise, the sequential connections of these wider layers result in a quadratic increase of computation and parameters, making the network inefficient and prone to overfitting. As the sampling rate of the signals is controlled during runtime, the lengths of heartbeats, represented as $(K, 1)$ where the agent's action determines the K value during runtime, can be different.

CNN architectures are generally designed to work with incoming data of a fixed sampling rate, and changes in the sizes of their inputs cause substantial performance loss, unnecessary computations, or failure in operation. To handle these limitations, we have utilized the **dimension-adaptive pooling (DAP)** layer [25] that makes DNNs flexible and more robust to changes in sampling rate. DAP layers addresses the aforementioned limitations without enforcing assumptions on the CNN architecture to be used. By building upon global and pyramid-pooling ideas, it maps the output dimensions of the last feature extractor layer into data whose dimensions match the expected dimensions of the fully connected layer. The functionality of the DAP layer in our problem can be explained as follows: Let (W, H) be the specified hyper-parameters for the pooling, where W and H represent the width and height of the window. In our case, since the input is a 1D ECG signal, it can be represented as $(W, 1)$. DAP first calculates the pooling parameters for the received input segments with different lengths, depending on the agent's decision. Then, it chooses the maximum pooling value to create outputs at a fixed dimension before the fully connected layer. Our readers can find a detailed explanation of the functionality of this layer in Reference [25].

After each convolutional layer, we applied batch normalization [19] and/or a rectified linear activation. The "+" and "−" signs near the convolutional layer show whether these operations are applied. We also used Dropout [38] before the last dense layer with a probability of 0.5 to prevent overfitting. The final fully connected softmax layer produces a distribution over the three output classes.

4 EXPERIMENTAL SETUP

4.1 Training CNN Classifier

We use data from the MIT-BIH Arrhythmia dataset [13, 29], which contains 48 half-hours of two-channel ambulatory ECG recordings, digitized at 360 samples per second, obtained from 47 subjects. Only the lead-II ECG signals are used for experiments for the MIT-BIH dataset. For a fair comparison with published results, we follow the evaluation settings that were most frequent in the state-of-the-art works. We have excluded the four paced records (102, 104, 107, 217) from the MIT-BIH dataset [29]. ECG beats in 22 recordings from the MIT-BIH dataset are included in the training set. Additionally, we have followed the subject-specific classifier scheme [17, 20, 22] where the training data consisted of two parts, a common part and a subject-specific part. The

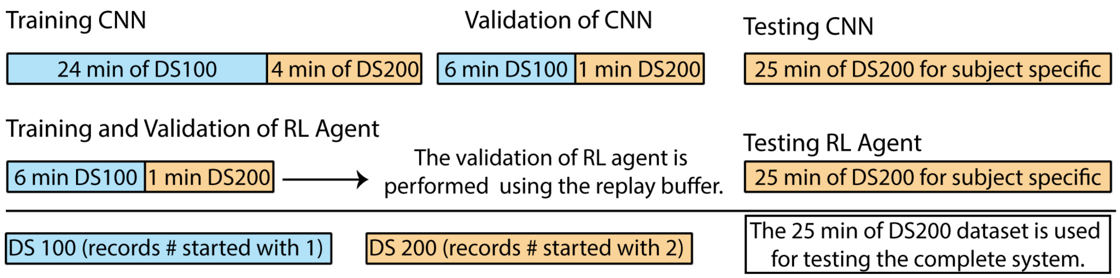

Fig. 5. The diagram showing how the training, validation, and test sets are curated.

common part of training data was selected from the first group (record # started with 1, also called DS100) and is used for all testing subjects (second record group, record # started with 2, also called DS200). Following the AAMI recommended practice, at most 5 minutes of recordings from a subject were used for classifier training purposes. So, the subject-specific part of training data included the heartbeats from the first 5 minutes of the ECG recording of each testing subject. The remaining 25 minutes of all records were used for testing. The details of training, validation and testing are illustrated in Figure 5.

Moreover, since the length of each heartbeat is controlled during runtime, the CNN classifier should adapt to the changes in the sampling rate. Therefore, we have used adaptive dimension training, which comprises dimension randomization and optimization with accumulated gradients as in Reference [25]. This process works by training the CNN on input data of several randomly selected dimensions (sampling rates). In this way, the model can learn the waveform morphological features of different sizes of heartbeats.

The classifier network is trained with Glorot initialization of the weights [12]. L2-regularization with 0.0002 is applied for each convolution operation for Inception, while the last linear layers are trained with a 0.00005 L2 value. We used the Adam optimizer [21] with the default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a mini-batch size of 80. The learning rate is initialized to 0.001.

While splitting the training data for validation, the first six minutes of the global training data (DS100) and the first minute of the local training data (DS200) are used; the remaining 29 minutes of DS100 and the 4 minutes DS200 are used for training. The training continues until 500 successive epochs, and the best model is chosen as the highest F1 score on the validation data. In general, the hyper-parameters of the network architecture and optimization algorithm were chosen by manual tuning. We essentially searched over the number of convolutional layers, the size and number of the convolutional filters for the architecture optimization.

4.2 Training Agent

The validation data of the machine learning model, which is the first six minutes of the global training data (DS100) and the first minute of the local training data (DS200), is used for training the agent. It is observed that if the same training data that is employed before for training the ML model is also used for the RL model, the agent becomes highly biased, since the reward function of the agent depends on the correct classification of the current heartbeat and the trained model has already seen those data before in the training phase. To prevent this biasing, the validation data of the machine learning model is used to create an environment for training the RL agent. Additionally, since the temporal relationships play a crucial role in the RL agent performance, we did not change the order of the beats for training, validation, and testing. Therefore, the agent observed transitions from N to SVEB/VEB and vice versa in the same record.

The network that is used for reinforcement learning was trained using random initialization of the weights with unit standard deviation. We used the Adam optimizer [21] with the default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a replay memory of size 10^6, replay-batch size of 8, and ϵ-greedy for exploration with ϵ decaying from 1

Table 1. Performance Comparison of Proposed Method with Related Works

Work/Classes	VEB				SVEB			
	Acc	Sen	Spe	Ppr	Acc	Sen	Spe	Ppr
Jiang and Kong [20]	98.1	86.6	**99.3**	**93.3**	96.6	50.6	98.8	67.9
Ince et al. [17]	97.6	83.4	98.1	87.4	96.1	62.1	98.5	56.7
Kiranyaz [22]	98.6	95	98.1	89.5	96.4	64.6	98.6	62.1
Zhai and Tin [51]	98.6	93.8	99.2	92.4	**97.5**	**76.8**	98.7	74.0
Ours (Baseline−Fs)	98.9	96.1	99.2	93.2	96.5	66.5	98.1	61.1
Ours (Baseline−Fs/2)	**99.0**	**96.3**	99.2	93.1	95.6	67.3	97.2	62.1
Ours (Baseline−Fs/4)	96.3	97.8	95.3	84.3	92.4	58.5	91.7	59.5
Ours (Baseline−Fs/8)	93.7	95.6	93.1	82.5	91.1	54.3	89.5	57.2
Ours (Baseline−Fs/16)	92.5	93.3	91.2	81.3	89.3	52.1	88.3	54.1
Ours (Adaptive)	97.6	92.1	98.2	85.2	97.2	62.1	**98.9**	**75.2**

to 0.01. The reward discount factor (γ) is set to 0.99. The target network is updated every 100 steps, while the RL agent is trained for 6 iterations over the training data.

5 EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Performance Evaluation of Adaptive CNN

Classification performance is measured using the four standard metrics found in the literature [22]: **classification accuracy (Acc), sensitivity (Sen), specificity (Spe)**, and **positive predictivity (Ppr)**. While accuracy measures the overall system performance, the other metrics are specific to each class, and they measure the ability of the classification algorithm to differentiate certain events from nonevents. The respective definitions of these four standard metrics using **true positive (TP), true negative (TN), false positive (FP)**, and **false negative (FN)** are as follows: Accuracy is the ratio of the number of correctly classified beats to the total number of beats classified, Acc = (TP+TN)/(TP+TN+FP+FN); Sensitivity is the rate of correctly classified events among all events, Sen = TP/(TP+FN); Specificity is the rate of correctly classified nonevents among all nonevents, Spe = TN/(TN+FP); and Positive Predictivity is the rate of correctly classified events in all detected events, Ppr = TP/(TP+FP). While comparing the proposed method with the related works, specifically, we only compare to studies in which the proposed classifier is trained only once using the combination of global training data and 5 minutes of each test patient rather than training a model for every patient individually [34, 45, 46]. Also, to make a fair comparison with the related works, we have explicitly calculated and indicated specific performances for **supraventricular ectopic beats (SVEB)** and **ventricular ectopic beats (VEB)**, as shown below.

Table 1 shows the performance of our proposed method in comparison to several other state-of-the-art works that follow the AAMI recommended practice. We have included different baselines in Table 1, where each is trained on the decimated version of the original sampling rate (360 Hz for MIT-BIH dataset) indicated by (Fs/x), while the adaptive one changes the size of the beats according to the decision made by the agent during runtime. The results, as shown in Table 1, indicate that the adaptive classifier puts more emphasis on the detection of the SVEB class, since its positive predictive and specificity outperform all the related works. Even the percentage difference achieves approximately 20%. However, it should be noted that the tradeoff between VEB and SVEB detection performance becomes apparent when the adaptive classifier is used. A closer inspection of the table shows that although the detection for the SVEB increases with adaptive classifier, the **sensitivity (Sen)** and **positive predictive (Ppr)** values of VEB beat type decreases by 4%. In addition, we observe similar behavior of our baseline to the architecture used in Section 1.1, where the performance of our baseline increases when trained with Fs/2. However, if the sampling rate is reduced significantly, such as by 8 and 16, then the performance degradation is more significant. The decrease in the number of calculations, which is the reduction of FLOPs,

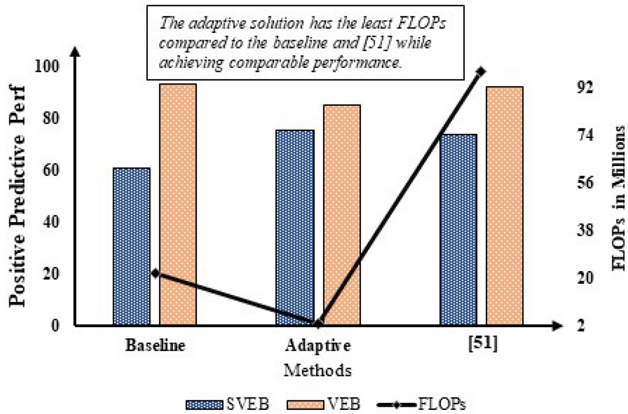

Fig. 6. Comparison of computational complexity and performance.

is not apparent in this table; therefore, we have compared the detection ratio of different classes with average FLOPs for architectures in Figure 6. As a metric for detection performance, we have used positive predictive value for different beat types similar to Section 1.1.

While comparing the FLOPs, we have only considered the architectures that are given explicitly (i.e., number of layers with kernel sizes) and we followed the FLOPs computation mentioned in Section 3.2. Furthermore, any preprocessing such as Fourier or wavelet transforms are ignored, and only the CNN architecture for classification FLOPs is calculated, as explained in Section 3.2. As shown in Figure 6, the adaptive solution reduces the overall FLOPs value by 30×, and 7.3× compared to Reference [51] and baseline where the adaptive sampling rate is not utilized. Furthermore, the adaptive solution achieves a reduction in overall computation while maintaining performance at a high level. This is exemplified by the SVEB beat type, where the positive predictive value experiences a notable increase of 10%.

Although the adaptive sampling rate decreases the overall computation while achieving comparable classification performance, it should be emphasized that one of the more significant contributions to emerge from this study is that our proposed method is not a substitute for other methods concerning resource-constrained devices; instead, it is a complementary method that can be used along with them. For example, pruning and quantization of deep learning models are widely used in literature [37, 44]; once those dynamic compression techniques have compressed a network, our adaptive sampling can still be applied to the ECG signals and fed to the compressed network. Or, different deep learning architectures that are concerned about energy and memory are recently proposed for ECG beat classification [34, 45]; our proposed method can be utilized with these architectures as an additional preprocessing step.

5.2 Ablation Studies

5.2.1 Architecture Comparison. As previously noted, our proposed deep learning architecture has two convolutional blocks with an inception block. We perform an ablation study of the network by removing each of these components individually and changing the number of kernels with size. To evaluate these components, we perform the same evaluation scheme described in Section 4.

For the first two cases of the ablation study, we change the kernel size of the convolutional filters in the Inception block. The first experiment $\{1, 3, 5\} \leftarrow \{1, 4, 16\}$ represents the decrease in kernel sizes from 1, 4, 16 to 1, 3, and 5. While in the second case, we increase the kernel size to 1, 8, and 64. The third case shows the performance when we exclude the Inception block completely from the architecture. And, in the last case, we decrease the kernel size of the 1st Conv. Block from 9 to 7. All the ablation study is performed when the method

Table 2. Performance Comparison of the Proposed Architecture and Its Ablation Variations across VEB and SVEB

Architecture	VEB				SVEB			
	Acc	Sen	Spe	Ppr	Acc	Sen	Spe	Ppr
Ours (Incept. Block Kernel width {1, 3, 5} ← {1, 4, 16})	92.3	86.6	94.6	83.2	91.1	53.4	84.4	58.9
Ours (Incept. Block Kernel width {1, 8, 64} ← {1, 4, 16})	95.4	90.7	98.1	85.1	96.3	60.1	98.3	72.7
Ours (w/o Inception)	91.2	83.3	93.1	81.5	89.1	51.5	86.4	58.0
Ours (1st Conv. Layer Kernel width 7 ← 9)	95.2	91.7	98.1	84.5	96.3	61.0	98.3	74.3
Ours (Baseline–Fs)	**98.9**	**96.1**	99.2	93.2	96.5	66.5	98.1	61.1
Ours (Adaptive)	97.6	92.1	98.2	85.2	97.2	62.1	**98.9**	**75.2**

Table 3. Performance Comparison of the Proposed RL and Contextual Bandits across VEB and SVEB

Modeling Method	VEB				SVEB			
	Acc	Sen	Spe	Ppr	Acc	Sen	Spe	Ppr
Ours (Contextual Bandits)	97.6	91.3	98.2	85.7	97.0	54.2	98.1	74.1
Ours (Baseline–Fs)	**98.9**	**96.1**	99.2	93.2	96.5	66.5	98.1	61.1
Ours (Adaptive)	97.6	92.1	98.2	85.2	**97.2**	62.1	**98.9**	**75.2**

is *Adaptive*. Table 2 shows the performance of the classifier decreases as the kernel size of convolutional layers decreases significantly. These results are likely to be related to dynamic change of beats during runtime. Since the combination of varying-length filters can capture different features compared to only the shorter-length implementation, the performance of the classifier heavily depends on the Inception block.

Contextual bandits as a control mechanism. In this case study, we examine a separate class of algorithms for solving sequential decision problems. Specifically, we chose the variant of the multi-armed bandit algorithm, the **contextual bandits (CB)**. The CB problem is a sequential decision-making problem where an agent chooses one of the possible actions (arms) based on the input context and receives a reward corresponding to the selected arm. The main difference between CB and RL is that the former assumes its actions do not affect the states and receives immediate rewards, whereas the latter assumes actions affect the states and the reward is the sum of future discounted rewards taking into consideration potential actions taken in the future given that it follows an optimal policy. Following the work in Reference [8], we apply the neural contextual bandits with Thompson sampling and consider the problem of adaptive sampling problem as a contextual K-armed bandit problem, where each arm represents the decimation actions and we have a finite number of rounds T. At every round $t \in [T]$, the agent observes K contextual vectors of size d $\{\mathbf{x}_{t,k} \in \mathbb{R}^d | k \in K\}$. When the agent selects an arm a_t it receives a corresponding reward r_{t,a_t}. The goal is to maximize the total expected reward or in other words to minimize the sum of regrets:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}(r_{t,a_t^*} - r_{t,a_t})\right], \tag{6}$$

where a_t^* is the optimal arm at round t that gives the maximum expected reward. CB takes the same ECG beats as RL as the contextual input. Our experiments show that RL outperforms CB, as shown in Table 3. Since the decimation action changes the ECG beat morphology that is used to classify the signal, RL is able to capture the contextual and temporal correlations existing in the consecutive ECG segments.

The contextual bandit neural network is a single hidden layer network with an input size equal to K arms times 256 features, 128 hidden units, and 1 output node to approximate the mean reward. The input contextual vector is constructed following Reference [52] such that each arm sees a corresponding vector in the form $\{\mathbf{x}; \mathbf{0}; \cdots; \mathbf{0};\}$ for arm 0, $\{\mathbf{0}; \mathbf{x}; \cdots; \mathbf{0};\}$ for arm 1, up to K where x is the feature vector. We set ν and λ to 1e^{-6} and 0.1. Each

Table 4. Performance Comparison of the Proposed Methodology on Different Experimental Setups

Method–Experimental setup	VEB				SVEB			
	Acc	Sen	Spe	Ppr	Acc	Sen	Spe	Ppr
Ours (Baseline–Fs)–CV	**99.2**	97.5	**99.7**	**98.7**	98.4	30.1	99.1	29.4
Ours (Adaptive)–CV	98.1	**98.3**	98.1	93.5	**98.5**	**40.0**	**99.2**	**33.5**
Ours (Baseline–Fs)	**98.9**	**96.1**	99.2	93.2	96.5	66.5	98.1	61.1
Ours (Adaptive)	97.6	92.1	98.2	85.2	97.2	62.1	**98.9**	**75.2**

context is trained for 100 iterations of stochastic gradient descent with a learning rate of 0.01 and weight decay $\lambda/counter$ where the counter increases by 1 up until the total training samples.

Performance of RL in different experimental setups. To observe the performance of reinforcement learning with different data sizes, we conducted an ablation study by changing the experimental setup. We performed 11 **cross-validations (CV)** in the MIT-BIH dataset, where we use records from four subjects to test such that at the end of the cross-validations, all subjects are used for testing. Although cross-validation is not a common evaluation scheme in the MIT-BIH due to heavily unbalanced beat distributions across subjects, we believe that this ablation study shows the performance of the proposed methodology under different experimental setups. For validation of the classifier and training of the RL model, we use four randomly selected subjects. The overall mean results are given in Table 4.

Table 4 shows that the classifier performance decreases significantly when the subject-specific training is not performed. However, our adaptive solution still performs better than the baseline while decreasing the overall energy consumption by ≈ 3–4 times. These results suggest that our proposed methodology outperforms or reaches the baseline models while being more energy-efficient at different experimental setups.

5.2.2 Effect of Downsampling. During our ablation studies, we also investigated the performance of the classifier for different classes, especially VEB and SVEB. As can be seen from Table 1, the PPR of VEB drops by 8% for the adaptive case compared to the baseline, which shows that downsampling produces beat patterns that are harder for our adaptive model to classify. The difference between the Adaptive and Baseline cases is the downsampled inputs to the classifier during training and inference. Therefore, first, we examined the frequency contents of the original input samples to analyze the effect of downsampling. We take the Fourier transform of original samples and calculate the ratio of **high-frequency (HF)** components in the whole spectrum, as shown in Equation (7):

$$K_{HP} = \frac{\sum_{k=25}^{K=f_s/2\ Hz} \tilde{X}[k]}{\sum_{k=0}^{K=f_s/2\ Hz} \tilde{X}[k]}, \tag{7}$$

where $X[k]$ is the Fourier transform of the samples, k is the frequency bins, and K_{HP} is the obtained ratio. After calculating the frequency ratio, we show the distribution using the histogram as in Figure 7.

As can be seen from Figure 7, the frequency ratio of VEB samples is more spread compared to SVEB. In other words, VEB samples are more diverse in terms of frequency contents. During downsampling, the frequency contents of samples will spread more, as shown in Equation (13).

$$X_{d_1}(e^{jw}) = \sum_{k=-\infty}^{\infty} x_{d_1}[k]e^{-jwk} \quad \text{and} \quad X_{d_2}(e^{jw}) = \sum_{k=-\infty}^{\infty} x_{d_2}[k]e^{-jwk}, \tag{8}$$

$$\text{where} \quad x_c(t) \xrightarrow[Sampling]{} x_{d_1}[nT] \xrightarrow[Decimation]{} x_{d_2}[nTa_t] \quad \text{with a sampling rate } T \tag{9}$$

$$X_{d_2}(e^{jw}) = \sum_{k=-\infty}^{\infty} x_{d_2}[k]e^{-jwk} = \sum_{k=-\infty}^{\infty} x_{d_1}[ka_t]e^{-jwk} \tag{10}$$

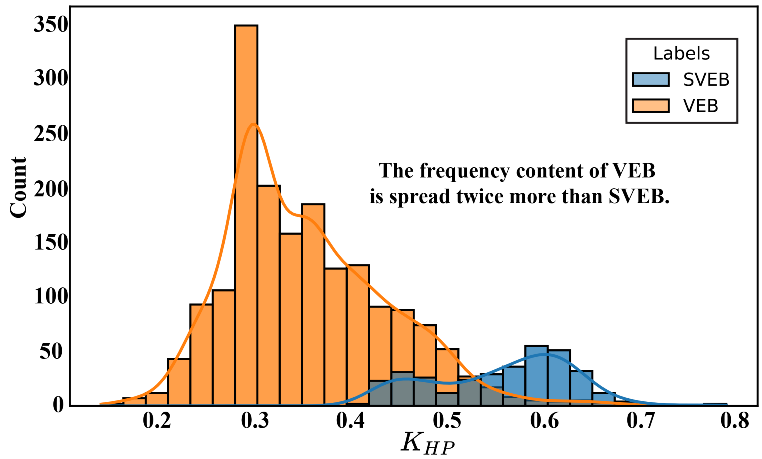

Fig. 7. Sample distributions in MIT-BIH according to the ratio of high-frequency components in the whole spectrum K_{HP}, for VEB and SVEB.

$$X_{d_2}(e^{jw}) = \sum_{\substack{k=\text{integer}\\ \text{multiple of}\\ a_1}} x_{d_1}[k]e^{-jwk/a_t} \tag{11}$$

$$X_{d_2}(e^{jw}) = \sum_{k_1=-\infty}^{\infty} x_{d_1}[k]e^{-jwk/a_t} \quad \text{since} \quad x_{d_1}[k] = 0 \quad \text{when} \quad \left\{\frac{k}{a_t}\right\} \notin \mathbb{Z} \tag{12}$$

$$X_{d_2}(e^{jw}) = X_{d_1}(e^{jw/a_t}). \tag{13}$$

Since the spreading increases when the downsampling rate a_t increases, features of samples in the frequency domain overlap more, making it harder for the model to learn the discriminative features. This can also explain why there is a performance increase in SVEB samples for the adaptive case compared to the baseline. When we investigate the sample distribution in the MIT-BIH dataset, we can see that the frequency content of SVEB samples is more in the high-frequency band. When we downsample the heartbeats, the samples from all classes will spread more to the high frequency as shown above, which eventually makes the classifier converge to a point to choose SVEB classes for most of the test samples. Another possible explanation for the performance degradation in VEB class is that when we look at Figure 7, we can see that the VEB samples have more features in the low band of the spectrum. Thus, when we downsample with a high rate, they will spread to the high-frequency band as in Equation (13), distorting the features of the class. This also proves that adaptive training tries to find common features in the same class across different decimation ratios. During our experiments, we show that if the decimation ratio is chosen correctly by the RL agent, then the performance-energy tradeoff can be optimized even though decimation results in loss of information.

6 MEMORY AND ENERGY CONSUMPTION EVALUATION

We evaluate our proposed method memory footprint and energy consumption on the EFM32 Giant Gecko ARM Cortex-M3-based 32-bit **microcontrollers (MCUs)**, which has a 1024 kB flash and 128 kB of RAM with a CPU speeds up to 48 MHz. Table 5 shows the execution time, energy consumption, and required memory for each operation that runs on the edge device. The operations are implemented and deployed to the target device using MATLAB (MATLAB and Coder Toolbox Release R2020b, the MathWorks, Inc).

Table 5. Memory Footprint, Execution Time, and Energy Consumption Evaluation on EFM32 Giant Gecko Development Board

Operations	Exe. Time (ms)	Avg. Energy (μJ)	Flash Memory Footprint (KB)	RAM Memory Footprint (KB)
Pre-processing	1,200	580.2	9.4	20.7
RL Agent	49.24	2.27	44.4	29.3
Overall	1,249	582	≤64 KB	≤32 KB

The overall execution time for a 5-second ECG segment takes $1,249$ ms in the edge device with 45.8 mW power consumption. As the pre-processing operation includes filtering, peak detection, segmentation, and normalization, its computational overhead dominates the overall operations. Also, our proposed method is compatible with any device with a minimum RAM of 32 KB. As a result, our method guarantees high performance while maintaining the low-power wearable devices' requirements of being resource-efficient in terms of energy and memory.

It is apparent from Table 5 that additional energy and time consumption due to the RL agent for choosing the sampling rate can be ignored compared to the overall execution time and energy consumption. A quantitative runtime analysis shows that the execution time of RL agent is 3.94% of the overall system. This overhead is much smaller if the energy consumption of the complete algorithm (pre-processing and RL agent) is investigated. The average energy consumption of the RL agent is only 0.38% of the entire system.

7 DISCUSSION AND FUTURE WORK

This article presents a novel and energy-efficient adaptive runtime sampling method to classify **electrocardiogram (ECG)** signals into different heart rhythms. To evaluate our methodology's performance, we compare our approach with several state-of-the-art methods that evaluate their classification results on the same datasets with the same evaluation method. We show that our proposed methodology reaches or outperforms the current state-of-the-art works in terms of classification performance for three different classes while being energy- and memory-efficient. Also, our proposed approach is not a replacement for other methods concerning resource-constrained devices that are used for continuous monitoring of patients; instead, it is a supportive method that can be used together with them. However, questions remain about whether the proposed approach's performance is excellent despite these promising results. Therefore, it is important to evaluate the limitations of our methodology.

First, in our proposed method, the state, the agent's features, heavily depends on the R-peak detection performance, since the observed state is defined based on the waveform of the current heartbeat. It is observed that when the detected R-peaks are wrong, the decision performance decreases severely due to improper segmentation. Second, even though deep reinforcement learning with **Double Q-learning (DQL)** [42] algorithm is used for making decisions during runtime, the different types of RL can be investigated using different features as state values. Also, although we have used the observation of the previous action with the current ECG feature embeddings in the last hidden layer, it could be beneficial to add additional past information, since the temporal correlation of ECG beats extends to more than just the previous beat. For example, the long-range dependence among the cardiovascular states has been shown in the literature several times [14, 47], therefore, models that concern long-range dependence of physiological signals [15] can give better results. We believe that further research should be undertaken to investigate the different modeling systems to increase performance while decreasing energy consumption. Finally, the MIT-BIH [29] is the most commonly used ECG dataset in literature; most state-of-the-art works and our proposed method focus on identifying small numbers of cardiac abnormalities (VEB, SVEB) that can be insufficient to represent the complexity and difficulty of ECG signals. Therefore, we believe there is abundant room for further progress in controlling the sampling rate for various ECG beats. For example, this article showed that increasing the sampling rate has no direct correlation with classification performance for some beat types. In future investigations, it might be helpful to use more comprehensive datasets to explore this phenomenon.

8 CONCLUSION

In this article, we propose a methodology for real-time adaptive sampling of ECG signals on low-power resource-constrained medical devices in terms of memory and battery (e.g., wearable devices) using a deep reinforcement learning setup leveraging the Double Q-Learning. Moreover, we also presented a data-driven novel RL agent that uses the inherent temporal correlation between consecutive ECG signals with waveform morphology to determine up to which degree to decimate the ECG signal without sacrificing the classification performance. Evaluation on the MIT-BIH dataset shows that our proposed adaptive sampling solution requires 32 KB of RAM and achieves up to 7.3× energy efficiency in the overall dataset without sacrificing any classification performance.

REFERENCES

[1] 2019. *Heart Disease Facts*. Retrieved from https://www.cdc.gov/heartdisease/facts.htm.

[2] 2021. NHANES - National Health and Nutrition Examination Survey Homepage. Retrieved from https://www.cdc.gov/nchs/nhanes/index.htm.

[3] 2021. NVSS - Public Use Data File Documentation. Retrieved from https://www.cdc.gov/nchs/nvss/mortality_public_use_data.htm.

[4] Piotr Augustyniak. 2020. Adaptive sampling of the electrocardiogram based on generalized perceptual features. *Sensors* 20, 2 (2020). DOI: https://doi.org/10.3390/s20020373

[5] Abdulkadir Celik, Khaled N. Salama, and Ahmed M. Eltawil. 2022. The Internet of Bodies: A systematic survey on propagation characterization and channel modeling. *IEEE Internet Things J.* 9, 1 (2022), 321–345. DOI: https://doi.org/10.1109/JIOT.2021.3098028

[6] Kun-chih (Jimmy) Chen, Po-Chen Chien, Zi-Jie Gao, and Chi-Hsun Wu. 2021. A fast ECG diagnosis by using non-uniform spectral analysis and the artificial neural network. *ACM Trans. Comput. Healthc.* 2, 3 (July 2021). DOI: https://doi.org/10.1145/3453174

[7] Berken Utku Demirel, Islam Abdelsalam Bayoumy, and Mohammad Abdullah Al Faruque. 2021. Energy-efficient real-time heart monitoring on edge-fog-cloud internet-of-medical-things. *IEEE Internet Things J.* 9, 14 (2021). DOI: https://doi.org/10.1109/JIOT.2021.3138516

[8] Berken Utku Demirel, Luke Chen, and Mohammad Al Faruque. 2022. Neural contextual bandits based dynamic sensor selection for low-power body-area networks. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED'22)*. Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/3531437.3539713

[9] D. L. Donoho. 2006. Compressed sensing. *IEEE Trans. Inf. Theor.* 52, 4 (2006), 1289–1306. DOI: https://doi.org/10.1109/TIT.2006.871582

[10] Sandra B. Dunbar, Olga A. Khavjou, Tamilyn Bakas, Gail Hunt, Rebecca A. Kirch, Alyssa R. Leib, R. Sean Morrison, Diana C. Poehler, Veronique L. Roger, and Laurie P. Whitsel. 2018. Projected costs of informal caregiving for cardiovascular disease: 2015 to 2035: A policy statement from the American Heart Association. *Circulation* 137, 19 (2018), e558–e577. DOI: https://doi.org/10.1161/CIR.0000000000000570

[11] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. 2019. A guide to deep learning in healthcare. *Nat. Med.* 25, 1 (2019), 24–29.

[12] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 9)*, Yee Whye Teh and Mike Titterington (Eds.). JMLR Workshop and Conference Proceedings, Chia Laguna Resort, Sardinia, Italy, 249–256. Retrieved from http://proceedings.mlr.press/v9/glorot10a.html.

[13] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation* 101, 23 (2000), e215–e220. DOI: https://doi.org/10.1161/01.CIR.101.23.e215

[14] Ary L. Goldberger, Luis A. N. Amaral, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, C.-K. Peng, and H. Eugene Stanley. 2002. Fractal dynamics in physiology: Alterations with disease and aging. *Proc. Nat. Acad. Sci.* 99, suppl_1 (2002), 2466–2472. DOI: https://doi.org/10.1073/pnas.012579499

[15] Gaurav Gupta, Chenzhong Yin, Jyotirmoy V. Deshmukh, and Paul Bogdan. 2021. Non-Markovian reinforcement learning using fractional dynamics. In *Proceedings of the 60th IEEE Conference on Decision and Control (CDC)*. IEEE Press, 1542–1547. DOI: https://doi.org/10.1109/CDC45484.2021.9683076

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

[17] Turker Ince, Serkan Kiranyaz, and Moncef Gabbouj. 2009. A generic and robust system for automated patient-specific classification of ECG signals. *IEEE Trans. Biomed. Eng.* 56, 5 (2009), 1415–1426. DOI: https://doi.org/10.1109/TBME.2009.2013934

[18] GUSTO Angiographic Investigators. 1993. The effects of tissue plasminogen activator, streptokinase, or both on coronary-artery patency, ventricular function, and survival after acute myocardial infarction. *New Eng. J. Med.* 329, 22 (1993), 1615–1622.

[19] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167 [cs.LG]

[20] Wei Jiang and Seong G. Kong. 2007. Block-based neural networks for personalized ECG signal classification. *IEEE Trans. Neural Netw.* 18, 6 (2007), 1750–1761. DOI: https://doi.org/10.1109/TNN.2007.900239

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). Retrieved from http://arxiv.org/abs/1412.6980.

[22] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. 2016. Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Trans. Biomed. Eng.* 63, 3 (2016), 664–675. DOI: https://doi.org/10.1109/TBME.2015.2468589

[23] Changxin Lai et al. 2021. Optimal ECG-lead selection increases generalizability of deep learning on ECG abnormality classification. *Phil. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.* 379, 2212 (2021).

[24] Yaqiong Liu, Mugen Peng, Guochu Shou, Yudong Chen, and Siyu Chen. 2020. Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things. *IEEE Internet Things J.* 7, 8 (2020), 6722–6747. DOI: https://doi.org/10.1109/JIOT.2020.3004500

[25] Mohammad Malekzadeh, Richard Clegg, Andrea Cavallaro, and Hamed Haddadi. 2021. DANA: Dimension-adaptive neural architecture for multivariate sensor data. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 5, 3 (Sep. 2021). DOI: https://doi.org/10.1145/3478074

[26] Hossein Mamaghanian, Nadia Khaled, David Atienza, and Pierre Vandergheynst. 2011. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. *IEEE Trans. Biomed. Eng.* 58, 9 (2011), 2456–2466.

[27] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T. Dudley. 2018. Deep learning for healthcare: Review, opportunities and challenges. *Brief. Bioinform.* 19, 6 (2018), 1236–1246.

[28] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning Convolutional Neural Networks for Resource Efficient Inference. arXiv:1611.06440 [cs.LG]

[29] G. B. Moody and R. G. Mark. 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Eng. Med. Biol. Mag.* 20, 3 (2001), 45–50. DOI: https://doi.org/10.1109/51.932724

[30] Jinghao Niu, Yongqiang Tang, Zhengya Sun, and Wensheng Zhang. 2020. Inter-patient ECG classification with symbolic representations and multi-perspective convolutional neural networks. *IEEE J. Biomed. Health Inform.* 24, 5 (2020), 1321–1332. DOI: https://doi.org/10.1109/JBHI.2019.2942938

[31] Jiapu Pan and Willis J. Tompkins. 1985. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* 3 (1985), 230–236.

[32] Md. Abdur Rahman and M. Shamim Hossain. 2021. An Internet-of-Medical-Things-Enabled edge computing framework for tackling COVID-19. *IEEE Internet Things J.* 8, 21 (2021), 15847–15854. DOI: https://doi.org/10.1109/JIOT.2021.3051080

[33] Iñaki Romero, Bernard Grundlehner, Julien Penders, Jos Huisken, and Yahya H. Yassin. 2009. Low-power robust beat detection in ambulatory cardiac monitoring. In *Proceedings of the IEEE Biomedical Circuits and Systems Conference.* 249–252. DOI: https://doi.org/10.1109/BIOCAS.2009.5372036

[34] Saeed Saadatnejad, Mohammadhosein Oveisi, and Matin Hashemi. 2020. LSTM-based ECG classification for continuous monitoring on personal wearable devices. *IEEE J. Biomed. Health Inform.* 24, 2 (2020), 515–523. DOI: https://doi.org/10.1109/JBHI.2019.2911367

[35] Rituparna Saha, Sudip Misra, and Pallav Kumar Deb. 2021. FogFL: Fog-assisted federated learning for resource-constrained IoT devices. *IEEE Internet Things J.* 8, 10 (2021), 8456–8463. DOI: https://doi.org/10.1109/JIOT.2020.3046509

[36] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE Internet Things J.* 3, 5 (2016), 637–646. DOI: https://doi.org/10.1109/JIOT.2016.2579198

[37] Moran Shkolnik, Brian Chmiel, Ron Banner, Gil Shomron, Yury Nahshan, Alex Bronstein, and Uri Weiser. 2020. Robust quantization: One model to rule them all. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 5308–5317. Retrieved from https://proceedings.neurips.cc/paper/2020/file/3948ead63a9f2944218de038d8934305-Paper.pdf.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 1929–1958.

[39] Ting-Wei Sun, Danish Ali, and Ayeu (Andy) Wu. 2022. Compressed-domain ECG-based biometric user identification using task-driven dictionary learning. *ACM Trans. Comput. Healthc.* 3, 3 (Apr. 2022). DOI: https://doi.org/10.1145/3461701

[40] Weifang Sun, Nianyin Zeng, and Yuchao He. 2019. Morphological arrhythmia automated diagnosis method using gray-level co-occurrence matrix enhanced convolutional neural network. *IEEE Access* 7 (2019), 67123–67129. DOI: https://doi.org/10.1109/ACCESS.2019.2918361

[41] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15).* 1–9. DOI: https://doi.org/10.1109/CVPR.2015.7298594

[42] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. arXiv:1509.06461 [cs.LG]

[43] S. Virani et al. 2020. Heart disease and stroke statistics—2020 update: A report from the American Heart Association. *Circulation* 141, 9 (2020), e139–e596. DOI: https://doi.org/10.1161/CIR.0000000000000757

[44] Wenxiao Wang, Minghao Chen, Shuai Zhao, Long Chen, Jinming Hu, Haifeng Liu, Deng Cai, Xiaofei He, and Wei Liu. 2021. Accelerate CNNs from three dimensions: A comprehensive pruning framework. In *Proceedings of the International Conference on Machine Learning (ICML'21)*.

[45] Jianbiao Xiao, Jiahao Liu, Huanqi Yang, Qingsong Liu, Ning Wang, Zhen Zhu, Yulong Chen, Yu Long, Liang Chang, Liang Zhou, and Jun Zhou. 2021. ULECGNet: An ultra-lightweight end-to-end ECG classification neural network. *IEEE J. Biomed. Health Inform.* 26, 1 (2021), 1–1. DOI: https://doi.org/10.1109/JBHI.2021.3090421

[46] Sean Shensheng Xu, Man-Wai Mak, and Chi-Chung Cheung. 2020. I-vector-based patient adaptation of deep neural networks for automatic heartbeat classification. *IEEE J. Biomed. Health Inform.* 24, 3 (2020), 717–727. DOI: https://doi.org/10.1109/JBHI.2019.2919732

[47] Yuankun Xue and Paul Bogdan. 2017. Constructing compact causal mathematical models for complex dynamics. In *Proceedings of the 8th International Conference on Cyber-Physical Systems (ICCPS'17)*. Association for Computing Machinery, New York, NY, 97–107. DOI: https://doi.org/10.1145/3055004.3055017

[48] Tsung-Yen Yang, Michael Hu, Yinlam Chow, Peter J. Ramadge, and Karthik Narasimhan. 2020. Safe reinforcement learning with natural language constraints. *arXiv preprint arXiv:2010.05150* (2020).

[49] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. 2021. Reinforcement learning in healthcare: A survey. *ACM Comput. Surv.* 55, 1 5 (Nov. 2021). DOI: https://doi.org/10.1145/3477600

[50] Silvio Zanoli, Tomas Teijeiro, Fabio Montagna, and David Atienza. 2020. An event-based system for low-power ECG QRS complex detection. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'20)*. 258–263. DOI: https://doi.org/10.23919/DATE48585.2020.9116498

[51] Xiaolong Zhai and Chung Tin. 2018. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access* 6 (2018), 27465–27472. DOI: https://doi.org/10.1109/ACCESS.2018.2833841

[52] Weitong Zhang et al. 2021. Neural Thompson sampling. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*.

Received 6 June 2022; revised 31 January 2023; accepted 6 May 2023