Energy-Aware Design Methodology for Myocardial Infarction Detection on Low-Power Wearable Devices

Mohanad Odema, Nafiul Rashid, Mohammad Abdullah Al Faruque
Department of Electrical Engineering and Computer Science
University of California, Irvine, Irvine, California, USA
{modema, nafiulr, alfaruqu}@uci.edu

ABSTRACT

Myocardial Infarction (MI) is a heart disease that damages the heart muscle and requires immediate treatment. Its silent and recurrent nature necessitates real-time continuous monitoring of patients. Nowadays, wearable devices are smart enough to perform on-device processing of heartbeat segments and report any irregularities in them. However, the small form factor of wearable devices imposes resource constraints and requires energy-efficient solutions to satisfy them. In this paper, we propose a design methodology to automate the design space exploration of neural network architectures for MI detection. This methodology incorporates Neural Architecture Search (NAS) using Multi-Objective Bayesian Optimization (MOBO) to render Pareto optimal architectural models. These models minimize both detection error and energy consumption on the target device. The design space is inspired by Binary Convolutional Neural Networks (BCNNs) suited for mobile health applications with limited resources. The models' performance is validated using the PTB diagnostic ECG database from PhysioNet. Moreover, energy-related measurements are directly obtained from the target device in a typical hardware-in-the-loop fashion. Finally, we benchmark our models against other related works. One model exceeds state-of-the-art accuracy on wearable devices (reaching **91.22%**), whereas others trade off some accuracy to reduce their energy consumption (by a factor reaching **8.26**×).

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

Mobile Health, Myocardial Infarction, Wearable Devices, Multi-Objective Bayesian Optimization, Neural Architecture Search

1 INTRODUCTION

Over the past decade, heart disease has been one of the leading causes of death in the United States of America. Heart disease causes more than 600,000 deaths per year which represent 1 out of every 4 deaths [1]. Myocardial Infarction (MI), also known as a heart attack, is one of the fatal forms of heart disease. It occurs when the heart muscle lacks enough supply of blood and essential nutrients due to blocked arteries. The more time that passes without adequate treatment, the more damage that is done to the heart muscle. MI is also characterized by a recurrent nature as 25% of Americans suffering from heart attacks every year have already had a previous one [1]. Furthermore, one out of every 5 heart attacks is silent. This means that the victim is unaware of the attack [1].

All this justifies the necessity of continuous monitoring, especially for people who have already contracted MI. Electrocardiograms (ECGs) are capable of detecting heart problems and are used to monitor the patient's condition by recording the heart's electrical signals. Most of the monitoring takes place in a clinical environment with heavy medical equipment. However, the need for immediate treatment and the risks associated with silent heart attacks require real-time continuous monitoring. Hence, the routine check at the physician may not be enough to mitigate the consequences of MI on the patient's health. For this, medical wearable devices equipped with ECG monitoring capabilities are becoming the most prominent option for real-time monitoring of fatal heart diseases like MI.

The traditional approach requires wearable devices to send raw data to an intermediary (e.g., smartphone) via Bluetooth [13]. After that, the data is relayed to a centralized cloud where it is processed for physicians to keep track of the patient's health. However, this is not suitable for wearable devices as they spend a significant portion of their limited energy on transmitting the raw data. Furthermore, this data relaying scheme is prone to transmission delays. Therefore, an alternative approach is to equip the wearable devices with intelligence to facilitate on-device data processing. Then, only the final classification label needs to be relayed further. This approach can be referred to as Edge Computing.

To promote intelligence on wearable devices, machine learning models have been implemented to classify the patients' condition. Yet, these models need to maintain decent performance while meeting the tight energy and memory constraints of wearable devices. In this regard, improving performance is associated with the increased complexity of the models which, in turn, implies more memory utilization and energy consumption. Thus, the problem becomes how to design models that balance this trade-off, ensuring adequate performance while operating within constrained devices.

Most of the machine learning models used in MI detection literature [16, 17] work on the extracted features from ECG segments. This extra pre-processing stage can dramatically increase the execution time, and consequentially, energy consumption. In this regard, Convolutional Neural Networks (CNNs) [11] are better alternatives as they are fast and capable of extracting features through their inherent convolution process. However, their large size and working memory requirements may not be satisfied by the resource-constrained wearable devices. To resolve this, authors in [14] have designed a Binary Convolutional Neural Network (BCNN) [6] for

ASPDAC '21, January 18–21, 2021, Tokyo, Japan

ACM ISBN 978-1-4503-7999-1/21/01.
https://doi.org/10.1145/3394885.3431513

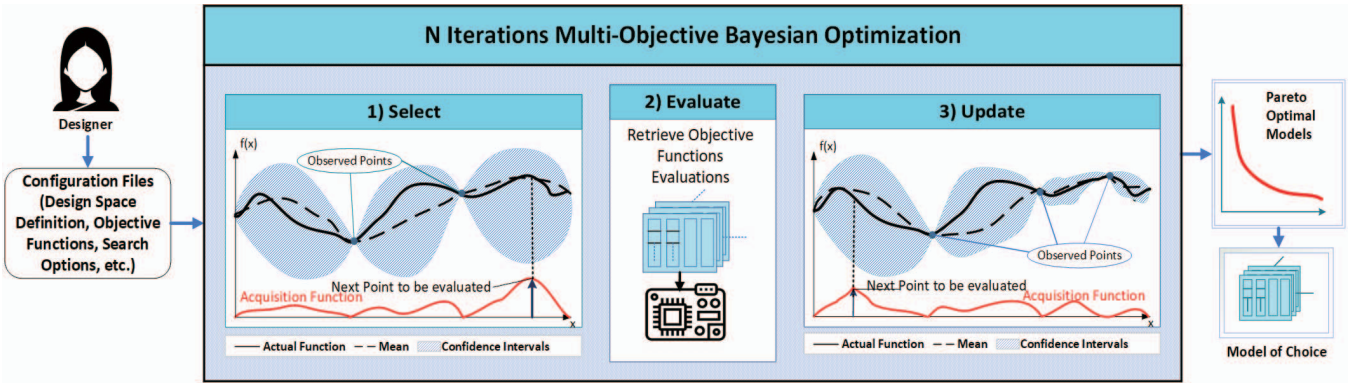

Figure 1: The design flow process using MOBO. The actual function is unknown in reality. Instead, a Gaussian Process (GP) model is constructed for each objective function and updated each iteration based on the information collected so far.

wearable devices. Their model design is characterized by having all of the model parameters represented in binary, leading to a significant reduction in the model's memory requirements. Moreover, they are energy efficient as they only perform binary operations throughout the network.

In this paper, we propose a methodology to incorporate Neural Architecture Search (NAS) [7] to co-optimize the design of BCNNs for MI detection with regard to accuracy and energy using Multi-Objective Bayesian Optimization (MOBO). Figure 1 shows the generalized design flow using our methodology where MOBO performs a systematic design space exploration of a BCNN-inspired search space to sample the most efficient models satisfying both objectives. Each sampled model is trained to estimate its accuracy before being deployed on the target device to retrieve the related energy measurements. The Bayesian models are updated each iteration with new data in order to improve the search strategy. Finally, our methodology renders a set of Pareto optimal neural architectures that represent the most suitable models for deployment on the target wearable devices. The main contributions of this paper are summarized as follows:

- A methodology is proposed to automate the design of BCNNs for MI detection on wearable devices through co-optimizing both accuracy and energy using real-hardware target device measurements.
- To the best of our knowledge, we are the first to propose a NAS-based design methodology working with time-series ECG signals.
- The performance of our methodology is validated using PTB diagnostic ECG database [4] from PhysioNet [9].
- In comparison with the state-of-the-art works for wearable devices, one of our explored models achieves the highest accuracy of **91.22%** while others achieve up to **8.26**× more energy efficiency.

2 RELATED WORK

2.1 Single Lead Wearable MI Detection

Many studies have been conducted aiming to achieve high performance on MI detection. However, most of them are targeting clinical setups or use multiple ECG leads, which are not suited for wearable devices. Here, we target studies that used only a single lead ECG signal for MI detection on wearable devices. The authors in [16] provided a wearable device solution using a Support Vector Machine (SVM). Their methodology incorporated a two-level classifier; the first-level classifier was computationally efficient while the more complex second-level classifier is only invoked when the first-level one fails to meet the classification confidence threshold. Features were extracted from the lead 11 ECG signal, and their SVM classifier achieved a 90% accuracy. In addition, the same authors proposed in [17] a hierarchical Random Forest (RF) classifier with multiple levels to achieve even more energy efficiency at the lower levels. Their 4-level full implementation achieves 83.26% accuracy, 87.95% sensitivity, and 78.82% specificity.

Those works rely heavily on the time-consuming and computationally expensive feature extraction process from the raw data. To overcome this, the authors in [3] implemented a 1-D CNN which directly classifies MI segments. They achieved an accuracy of 93.53% using lead 2 ECG data. However, their CNN architecture comprised 11 layers requiring a lot of intermediate calculations which would not suit the working memory limitations of most wearable devices. The authors in [14] addressed this issue by designing a BCNN which can operate under strict memory constraints. Their approach achieved a 90.29% accuracy while maintaining the energy and memory efficiencies. To the best of our knowledge, their work achieved the state-of-the-art result for MI detection on low-power wearable devices. Therefore, this work in [14] will represent the basis for our proposed methodology's design space.

2.2 Neural Architecture Search

Recently, NAS has been gaining significant attention as a systematic approach that can automate the design of neural networks. It aims to find optimal architectural designs that can outperform the handcrafted ones with respect to the design optimization objectives. In this regard, NAS incorporates a closed-loop cycle of selecting a single neural architecture each iteration for evaluation. Then based on these evaluations, the search strategy is updated to find more suitable neural architectures in the following iterations. NAS can be attained through various methods including reinforcement learning, evolutionary algorithms, gradient-based algorithms, or Bayesian optimization [7].

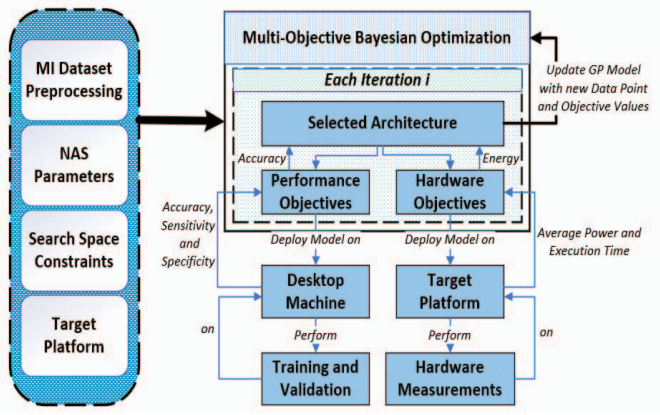

Figure 2: Our proposed energy-aware design methodology of neural architectures for MI detection on wearable devices.

In addition, hardware-aware optimization has been gaining interest over the past years. Authors in [18] use power and memory prediction models in their search process to prevent selecting model architectures that violate a predefined set of hardware constraints. Another work in [5] adopts MOBO-based NAS to co-optimize accuracy and energy using direct measurements from the target platforms. While our work is closely related to the one in [5], their approach does not address the same level of constrained wearable devices suited for mobile health applications.

The authors in [8] also utilized NAS for the same level of constrained devices. They proposed a framework combining NAS and network pruning to render architectures that can be deployed on an Arduino Uno (16 MHz clock, 32 KB flash and 2 KB RAM). To achieve this, their approach involved applying MOBO over a search space of various architectural parameters with the objective of minimizing error, working memory, and the model size for various image classification tasks. Our methodology is unique not only in the application context but also in the sense that MOBO is directed towards minimizing error and energy consumption. Furthermore, binarized models inspired by [6, 14] are utilized to satisfy the memory limitations of the target wearable devices.

3 OUR METHODOLOGY

3.1 Overview

The multi-objective optimization problem can be formulated as $\boldsymbol{min_{x \in X}(error(x), energy(x))}$, in which the goal is to find a network architecture parameterized by $\boldsymbol{x}$ from the search space $\boldsymbol{X}$ that minimizes two objective functions: MI detection error and energy consumption on the target device. As shown in Figure 2, our methodology does not assume direct closed-form models for both functions with respect to the neural architectural parameters. Instead, the problem is treated as a black box optimization one. This requires that for each sampled architecture, the accuracy loss and energy consumption should be evaluated to understand better how they relate to the architectural search parameters $\boldsymbol{x}$. While the classification loss is estimated computationally, energy is obtained through measuring power and execution time directly from the target device. However, as the two objectives are conflicting in nature, improving on one objective will negatively impact the other. Therefore, the outcome of this problem would have to be a set of Pareto optimal architectures $\boldsymbol{X^\star}$ which dominate all other explored architectures but not each other. Formally in a minimization context, a point $\boldsymbol{x}$ is said to dominate $\boldsymbol{x'}$ if for every objective function $\boldsymbol{f_k}$, $\boldsymbol{f_k(x) \leq f_k(x')\forall k}$ and at least one inequality is strict.

To solve this problem, we exploit MOBO [15] for our black-box optimization problem. Bayesian optimization methods provide efficient design space exploration in order to sample the most promising candidates that meet the minimization requirements of the objective functions. This is extremely useful when the search space is large and when evaluating an objective function is costly (like computing the loss function in our case). In this problem, once an architecture is sampled from the search space, the objective functions are evaluated to determine whether this candidate architecture should belong to the optimal set or not. Also, with each evaluation, the search strategy is updated to find better architectures in the following iterations.

3.2 Binary Convolutional Neural Network

Our search space is inspired by the BCNN architecture proposed in [14]. Their aim was to design an efficient CNN that can fit into wearable devices with limited memory while conserving energy resources. To achieve this, the model weights are limited only to +1 or -1. Moreover, only a binary activation function is used to clamp the inputs to either +1 or -1 as introduced in the binarized neural networks [6]. This binary representation of weights achieves 32× memory efficiency compared to the standard floating-point representation. Although the weights are in binary, temporaries generated between convolutional layers are still represented in floating-point. They require a lot of working memory resources which can still present an issue for wearable devices. To handle this, the computation order of inference in a binarized neural network has been modified following the work in [12]. Unlike in the traditional order, the resulting temporaries after the convolution layer are not stored in memory. Alternatively, they are directly passed to the pooling layer followed by batch normalization and binary activation layers. This makes the models not only memory efficient but also energy efficient because of the faster and less complex binary operations. Figure 3 shows the modified order of computation in one BCNN block for processing heartbeat segments.

3.3 Multi-Objective Bayesian Optimization

Given previous evaluations for each of the $\boldsymbol{k}$ objective functions $\boldsymbol{f_k(x)}$, the goal is to find samples that provide more information about the Pareto optimal set $\boldsymbol{X^\star}$. MOBO serves this purpose by performing a sequential design space exploration where each objective function is replaced by a surrogate, cheaper to evaluate, probabilistic Gaussian Process (GP) model. Let $\boldsymbol{D_n = \{(x_i, Y_i)\}_{i=1}^n}$ represent the set of all the queried points up to iteration $\boldsymbol{n}$; where for each step $\boldsymbol{i}$, $\boldsymbol{x_i}$ represents the sampled architecture at iteration $\boldsymbol{i}$ and $\boldsymbol{Y_i}$ represents its corresponding vector of $\boldsymbol{k}$ real evaluated values from the $\boldsymbol{k}$ objective functions. It is assumed that for each function $\boldsymbol{f_k(x)}$, evaluations $\mathbf{f_k} := \boldsymbol{Y_{1:n}[k]}$ are jointly Gaussian with mean $\boldsymbol{m}$ and co-variance $\boldsymbol{K}$, *i.e.*, $\mathbf{f_k}|\boldsymbol{x_{1:n}} \sim \boldsymbol{N(m, K)}$. This means each GP model at iteration $\boldsymbol{n}$ represents a distribution over all the possible functions of $\boldsymbol{f_k(x)}$ based on the data collected so far. This

Table 1: Ranges of the Architectural Search Parameters.

Parameter	Search Parameters Ranges
# BCNN blocks	*[1-3]*
# filters	*[2-5], [2-5], [2-5]*
Conv. layer kernel length	*[10-120], [10-70], [5-20]*
Conv. layer kernel stride	*[1-2], [1-2], [1-2]*
Pool. layer kernel length	*[2-3], [2-3], [2-3]*

distribution is known as the posterior, and it represents the current belief about the shape of functions that most likely fit D_n.

The next sample from the search space is selected using an acquisition function $\vartheta(x)$. The merit in using $\vartheta(x)$ is that, unlike the k objective functions, it is analytically available, making it much cheaper to evaluate than any $f_k(x)$. Hence for each iteration n, $\vartheta_n(x)$ is constructed using one of the k GP models to identify which point should be queried next. The GP model selected to construct $\vartheta_n(x)$ is chosen based on the improvement potential with regard to that specific objective function. Once the GP model is chosen, $\vartheta_n(x)$ is formulated to yield high values where the uncertainty of the probabilistic model is high (exploration), and around where the GP has had the best evaluations (exploitation). Then, the sample that maximizes $\vartheta_n(x)$ is selected to be the next query point x_{n+1}. In the following iteration $n+1$, the objective functions are evaluated yielding Y_{n+1}. Given this new data pair and the previous ones, the GP models are updated using the new dataset $D_{n+1} = D_n \cup (x_{n+1}, Y_{n+1})$. MOBO proceeds with this select-evaluate-update loop until the final iteration N is reached, and the Pareto set at that iteration is rendered as the final solution.

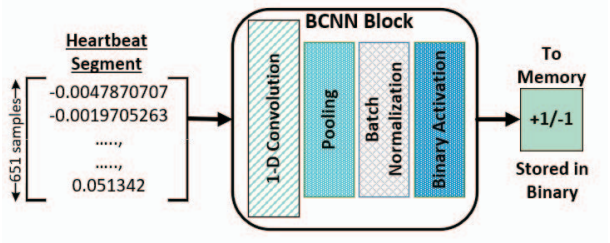

Figure 3: Processing heartbeat segments through the layers of the BCNN block and the final result is stored in binary.

4 EXPERIMENTAL SETUP

The multi-objective Bayesian optimization is built on top of Dragonfly [10]. We use Thompson sampling [**?**] for our acquisition function. Wrapper scripts are implemented around the objective functions to automate the selected models' generation, training, and deployment onto the target board. The details are provided below:

4.1 Training Process

Lead 11 ECG dataset from PTB diagnostic ECG database [4] is used for training and testing the models during and after the search process. It contains data for 200 subjects where 148 subjects suffer from MI, and the remaining 52 are normal. Out of the obtained heartbeat segments, 44214 segments are classified as MI while 6157 are normal. Since the number of MI segments are 7× the number of normal ones, we ensure proper training by dividing the segments into 7 groups. Each group will always contain all the normal segments combined with around 6316 MI segments. Then for each group, a 10 fold cross-validation is performed. In this scheme, each group is divided into 10 folds where for each fold, a unique 10% of that group's segments are used for testing while the remaining 90% are for training and validation. Each model selected during the search process is trained for 20 epochs with Adam optimizer, a learning rate of 0.007, and softmax cross-entropy as the loss function. For each group, the model's performance is averaged across all folds. Finally, the model's overall performance is estimated as the average across the entire 10 fold cross-validations in all groups.

4.2 Target Device

Our proposed design methodology targets low-power medical wearable devices like SmartCardia INYU [19]. This device was used by the related works in [16, 17]. It is equipped with an ultra-low-power Microcontroller STM32L151 running on an ARM Cortex-M3 with a maximum clock frequency of 32 MHz. The device also has a 48 KB RAM, 384 KB of Flash memory, and 710 mAh battery. The device also possess an ECG sensor to retrieve ECG signals through a single lead. For our experiments, we utilize both a desktop machine with a GeForce RTX 2070 SUPER and an EFM32 Leopard Gecko [2] as the low-power target device. The Bayesian search and the accuracy estimation procedures are performed on the desktop machine. Then to retrieve the relative hardware measurements, the model is converted into its corresponding C code implementation and automatically flashed onto the EFM32 board.

The EFM32 Leopard Gecko development board has been chosen for our experiments as it runs on the same ARM Cortex-M3 as SmartCardia INYU and has similar specifications. Estimating the energy consumption from hardware measurements can be detailed as follows: First, the execution time for a single inference of an ECG segment is calculated once the cycle count per inference is retrieved. After that, the target device is reset. Then, the average power over the calculated execution time for a single inference is measured. Finally, the energy consumption per inference can be computed directly by multiplying both the average power and the execution time.

5 RESULTS AND DISCUSSION

5.1 Experiments

We have performed multiple experiments to assess the effectiveness of MOBO within this problem context, as shown in Figure 4. The first experiment incorporated conducting MOBO over a BCNN-inspired search space. The chosen architectural search parameters are defined in Table 1, and their ranges are shown. Multiple ranges indicate the respective range of values for each consecutive BCNN block in an architecture. For convenience, this experiment was divided into 3 child experiments where for each one, the number of blocks was fixed as either one, two, or three to manage the dependency of the search parameters on the number of blocks. The search spaces for each child experiment accounted for 1.78×10^3, 1.73×10^6, and 4.44×10^8 possible architectures, respectively. Each child experiment was run for 200 iterations, and

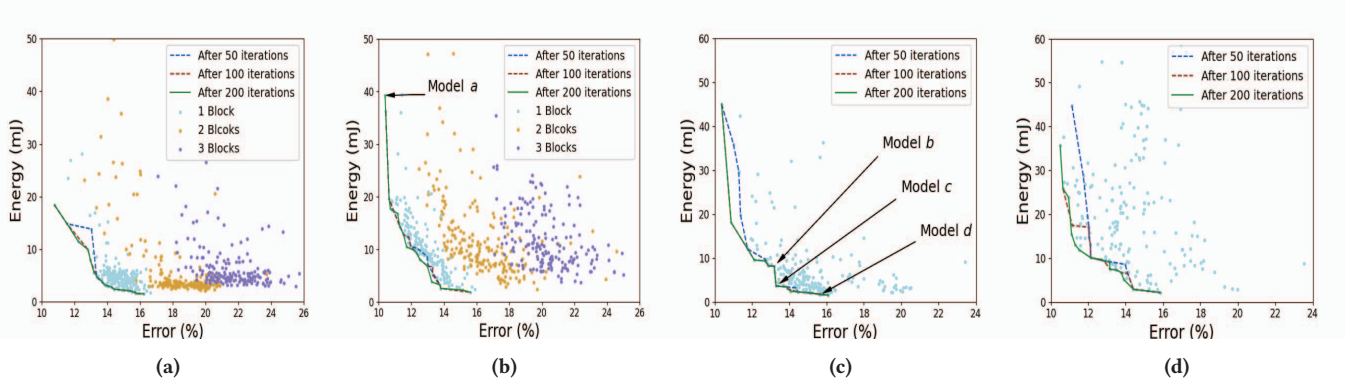

Figure 4: Results from our experiments. Sub-figures (a) and (b) show MOBO and normalized-MOBO over 3 reference architectures, respectively. While (c) and (d) compare MOBO and random sampling, respectively, over one block reference architecture.

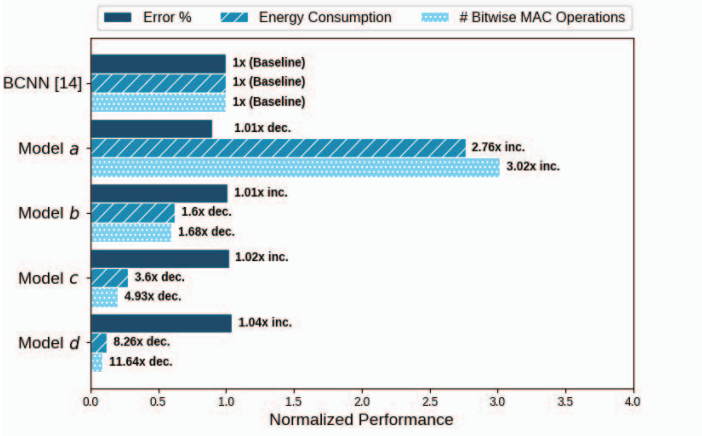

Figure 5: Analysis of Bitwise MAC Operations Count, Energy Consumption and Error for the Binary Based Models.

their combined results are shown in Figure 4a. The evolution of the ***combined*** Pareto frontier over 50, 100 and 200 iterations from each child experiment is shown. Two observations can be made here. The first is that MOBO tends to explore more around models that minimize energy consumption because the potential for improvement with respect to energy is greater than that with respect to error. The second observation is about how the single block architecture models dominate those from the other two architectures with respect to both objective functions.

Based on the first observation, the second experiment is designed to allow biasing the search in favor of one objective function over the other. Hence, rather than just directly using the real function evaluations, we add the option to normalize those evaluations in the Bayesian search process. This required modifying each function evaluation at every iteration n from $f_{kn}(x) := Y_n[k]$ to $f_{kn}(x) := \alpha_k \times \frac{Y_n[k]-min_k}{max_k-min_k}$, where α_k, min_k, and max_k are the bias constant, minimum, and maximum values of the k^{th} objective function, respectively. Since the first experiment was more biased towards energy, we set α_k for all objectives to 1 and use the *min* and *max* values from the previous experiment and re-run it. Figure 4b shows that the sampled architectures are more spread out than those in Figure 4a, indicating that MOBO has become more neutral in its search with respect to both objective functions.

The final experiment was to assess the effectiveness of the Bayesian search in terms of design space exploration. Based on the second observation from the first experiment, we re-run that non-normalized experiment twice but only for the one-block architecture. Bayesian search is used for the first run while the second employs random sampling. Figures 4c and 4d show their respective results. It can be observed that the Bayesian approach is much more systematic in its search to minimize the objective functions. This is evident through the rapid convergence of the Pareto frontier in the Bayesian experiment, as it is almost the same after 100 and 200 iterations.

Table 2: Models' Architectural Parameters

Model	# filters	Conv. len.	Conv. str.	Pool. len.
BCNN [14]	3	100	2	3
a	4	117	1	2
b	3	55	2	2
c	4	13	2	2
d	2	11	2	2

5.2 Final Benchmarking

The four models pointed out in Figures 4b and 4c are the ones we use for our final benchmarking. Their architectural search parameters values are presented in Table 2. Regarding their memory footprint, our **models *a***, ***b***, ***c***, and ***d*** use up around **19.33**, **19.12**, **19.17**, and **19.05** KB of flash and **3.69**, **3.54**, **3.63**, and **3.52** KB of RAM, respectively. This indicates that models from our design space comply with the low memory requirements of medical wearable devices like SmartCardia INYU. Next, we re-train those models to the full 100 epochs and compare them against the BCNN implementation in [14]. As shown in Figure 5, as the complexity of the model grows, so does the number of bitwise Multiply and Accumulate (MAC) operations. This, in turn, leads to increased energy consumption. However, as complexity is reduced, the energy savings are significant in comparison to the loss in accuracy. For instance, our model

Table 3: Comparison between Our Models and Previous Works with regard to Performance and Energy Metrics

	Accuracy (%)	Sensitivity (%)	Specificity (%)	Avg. Power (mW)		Exec. Time (ms)		Energy (mJ)	
Model				*14 MHz*	*48 MHz*	*14 MHz*	*48 MHz*	*14 MHz*	*48 MHz*
SVM [16]	90	-	-	14.24	46.92	13049.14	4303.28	185.82	201.91
RF [17]	83.26	87.95	78.82	14.34	46.98	13278.69	4378.69	190.42	205.71
BCNN [14]	90.29	90.41	90.16	14.52	46.71	893.14	279.86	12.97	13.07
Model *a*	**91.22**	**91.57**	**90.86**	**14.47**	**47.07**	**2477.77**	**846.39**	**35.85**	**39.84**
Model *b*	**89.63**	**90.01**	**89.24**	**14.63**	**46.97**	**553.68**	**176.74**	**8.10**	**8.30**
Model *c*	**88.26**	**87.27**	**89.27**	**15.30**	**47.08**	**235.2**	**80.54**	**3.60**	**3.79**
Model *d*	**86.92**	**85.91**	**87.96**	**15.31**	**47.14**	**104.30**	**36.23**	**1.57**	**1.71**

d incurs **1.04×** more detection error than the BCNN, yet it is **8.26×** more energy efficient.

Finally, we benchmark our retrained models against the SVM [16], RF [17], and BCNN [14] works. We compare their performance in terms of accuracy, sensitivity, and specificity metrics. Additionally, we also re-implement these works on the EFM32 board to ensure consistency of the energy consumption estimation across them all. However, it should be noted that although we report the best performance values for the SVM and RF, we only implement their first level classifiers for the energy-related evaluations. This is justifiable since the first level classifiers are the most efficient in terms of the execution time and energy consumption. The energy-related readings are measured at 14 MHz (default) and 48 MHz (maximum) operating frequencies of the EFM32 board for validation. Table 3 shows all measurements across all performance and energy metrics. Our **Model *a*** achieves the highest scores across the 3 performance metrics, whereas our remaining models are the most energy-efficient at the cost of some performance drop.

6 CONCLUSION

Adding intelligence to low-power wearable devices presents a design conundrum regarding the trade-off between high performance and energy efficiency. To address this, our proposed methodology provides a systematic automated design space exploration of efficient neural networks for MI detection on wearable devices. Our MOBO-based methodology allows for co-optimizing both detection error and energy consumption on the target device to render a Pareto optimal set of binarized models, allowing designers to choose their most suitable architectural design. Also, designers would be able to bias the search in the design process towards one objective or the other based on their preferences. To adhere to the memory limitations, our methodology explores the design space of variants of the BCNN architecture suitable for deployment on wearable devices. Experimental evaluation shows that one of our explored models achieves an accuracy of **91.22%**, outperforming the MI detection state-of-the-art performance on wearable devices. Other explored models trade off some accuracy to conserve more energy (as high as **8.26×**).

ACKNOWLEDGMENTS

This research was partially supported by NSF award CMMI 1739503. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect views of our funding agencies

REFERENCES

[1] 2020. *Deaths and Mortality.* Retrieved June, 2020 from https://www.cdc.gov/heartdisease/facts.htm
[2] 2020. *EFM32 Leopard Gecko|Silicon Labs.* Retrieved June, 2020 from https://www.silabs.com/products/mcu/32-bit/efm32-leopard-gecko
[3] U. Rajendra Acharya et al. 2017. Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Inf. Sci.* 415 (2017), 190–198.
[4] R. Bousseljot, D. Kreiseler, and A. Schnabel. 1995. Use of the PTB's ECG signal database CARDIODAT via the Internet. 40(s1) (1995), 317–318.
[5] Lile Cai, Anne-Maelle Barneche, Arthur Herbout, Chuan Sheng Foo, Jie Lin, Vijay Ramaseshan Chandrasekhar, and Mohamed M. Sabry Aly. 2019. TEA-DNN: The Quest for Time-Energy-Accuracy Co-optimized Deep Neural Networks. In *ISLPED 2019*. IEEE, 1–6.
[6] Matthieu Courbariaux and Yoshua Bengio. 2016. BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. *arXiv e-prints* (2016).
[7] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural Architecture Search: A Survey. *J. Mach. Learn. Res.* 20 (2019), 55:1–55:21.
[8] Igor Fedorov, Ryan P Adams, Matthew Mattina, and Paul Whatmough. 2019. SpArSe: Sparse Architecture Search for CNNs on Resource-Constrained Microcontrollers. In *Advances in Neural Information Processing Systems 32 NIPS 2019*. Curran Associates, Inc., 4977–4989.
[9] A. Goldberger, L.Amaral, L. Glass, J.Hausdorff, P. C. Ivanov, R. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. 101(23) (2000), e215–e220.
[10] Kirthevasan Kandasamy et al. 2020. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *Journal of Machine Learning Research* 21, 81 (2020), 1–27.
[11] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11 – 26.
[12] Bradley McDanel, Surat Teerapittayanon, and H. T. Kung. 2017. Embedded Binarized Neural Networks. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, EWSN 2017,*. 168–173.
[13] Nafiul Rashid, Manik Dautta, Peter Tseng, and Mohammad A. Al Faruque. 2020. HEAR: Fog-enabled Energy Aware Online Human Eating Activity Recognition. *IEEE Internet of Things Journal, doi:* (2020).
[14] Nafiul Rashid and Mohammad A. Al Faruque. 2020. Energy-efficient Real-time Myocardial Infarction Detection on Wearable Devices. In *42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC*.
[15] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* 104, 1 (2016), 148–175.
[16] Dionisije Sopic, Amin Aminifar, Amir Aminifar, and David Atienza. 2017. Real-time classification technique for early detection and prevention of myocardial infarction on wearable devices. In *IEEE Biomedical Circuits and Systems Conference, BioCAS 2017*. IEEE, 1–4.
[17] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza. 2018. Real-Time Event-Driven Classification Technique for Early Detection and Prevention of Myocardial Infarction on Wearable Systems. *IEEE Trans. Biomed. Circuits and Systems* 12, 5 (2018), 982–992.
[18] Dimitrios Stamoulis, Ermao Cai, Da-Cheng Juan, and Diana Marculescu. 2018. HyperPower: Power- and memory-constrained hyper-parameter optimization for neural networks. In *2018 Design, Automation & Test in Europe Conference & Exhibition DATE 2018*. IEEE, 19–24.
[19] Grégoire Surrel, Francisco J. Rincón, Srinivasan Murali, and David Atienza. 2015. Design of ultra-low-power smart wearable systems. In *16th Latin-American Test Symposium, LATS 2015*. IEEE Computer Society, 1–2.