Resource efficient personalized ECG beat classification via temporal logic synthesis

Jun Zhou
Department of Computer Science
National University of Singapore
zhoujun@comp.nus.edu.sg

Weng-Fai Wong
Department of Computer Science
National University of Singapore
wongwf@nus.edu.sg

***Abstract*—According to the World Health Organization, cardio-vascular diseases accounts for 31% of all deaths worldwide in 2016. Detecting the onset of heart irregularities can potentially save many lives. The ubiquity of wearable devices opens up the possibility of having a heart disease detection at everyone's disposal. To enable this, low energy ECG classification is needed. Unlike previous methods of using signal processing or even deep learning networks, this paper is the first to propose a low cost means to detect abnormal ECG beat signals by first synthesizing temporal logic formulas from training signals, and then checking if the synthesized formulas by the input signal at runtime. Our results show that the method has a high accuracy in detecting abnormal ECG beats while requiring significantly lower computation resource. Compared to It takes only a state-of-the-art convolutional neural network approach, our method achieves a comparable accuracy but with 0.3% of memory, and millions of computation operations, hence energy, saved.**

***Index Terms*—ECG, Classification, Temporal Logic**

I. Introduction

Cardio-vascular disease is a major cause of death worldwide. Electrocardiograph (ECG) records the electrical activities of the heart in the heartbeats using electrodes placed over the skin and reflects many kinds of heart irregularities. It is key to the detection as well as diagnosis of many heart ailments.

The ubiquity of wearable ECG devices such as Apple Watch 4 and Amazfit Verge 2 opens up the possibility of having a heart disease detector at everyone's disposal. To enable this, low energy ECG classification is needed. In this paper, we propose to synthesize *metric interval temporal logic* (MITL) formulas to detect abnormal ECG beats. MITL introduces temporal operators with time bounds in addition to propositional logic formulas to represent and reason about time. For example, $F[10, 20](1.0 \leq x \leq 2.0)$ means "between time points 10-20, eventually x has to reach the range $[1.0, 2.0]$". ECG signals can be checked against MITL formulas whose satisfiability can be used to classify the signals.

We define the problem as follows: Given a set of normal ECG beat signals from a user as the *training* set, synthesize a MITL formula such that the *testing* signals of normal beat (that are not found in the training set) satisfy the formula while abnormal ones do not.

This research is supported in part by Programmatic grant no. A1687b0033 from the Singapore government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain).

Temporal logic checking is a formal and efficient computation so it can be used for resource efficient ECG classification on wearable devices. Our results show that the method has a high accuracy in detecting abnormal ECG beats while requiring significantly lower computation resource. It takes only 0.3% of memory, and saves millions of computation operations, and hence energy, compared to a state-of-the-art convolutional neural network (CNN) approach.

II. Temporal Logic Synthesis

A. Metric Interval Temporal Logic

The *atomic* propositions for our setting are of the form $(\ell \leq x \leq u)$ where ℓ, u are real numbers. The proposition $(\ell \leq x \leq u)$ says "the current value of x lies in the interval $[\ell, u]$." We use a finite set of atomic propositions. Atomic propositions can be combined using the usual propositional logic operators $\neg$ ('not'), $\vee$ ('or') and $\wedge$ ('and') to form compound propositions. In addition, two temporal operators are used in conjunction with propositions to form logic formulas:

- $F[k, k']\psi$ (read as 'eventually ψ') holds if ψ holds at some time point t such that $k \leq t \leq k'$.
- $G[k, k']\psi$ (read as 'globally ψ') holds if ψ holds for all time points t such that $k \leq t \leq k'$.

where ψ may be an atomic proposition, a compound proposition, or (recursively) a logic formula involving F or G. The formal definition of MITL can be found in [1].

B. Method Overview

We use a two-phase workflow to synthesize the logic formulas. First, we generate the structures of the formula, which is the formula with the time and atomic proposition bounds as parameters, e.g. "$F[k, k'](\ell_1 \leq x \leq u_1)$". We then use simulated annealing to synthesize values of these parameters to obtain the concrete formula. From the above structure, a formula could be obtained as "$F[10, 20](1.0 \leq x \leq 2.0)$". The overview of the method is shown as the 1)-5) in Figure 1:

1) First, given a set of *ECG beats*, we automatically generate the *structure* of the formula according to the above assumptions.
2) According to the structure, the *value generator* generates concrete values for the parameters to produce the candidate formula.

3) We then check the candidate against the training ECG data and obtain a loss measure.
4) The loss measure is fed to the simulated annealing approach, which decides to terminate, or generate a new candidate.
5) After iterations of search, the simulated annealing report the optimal formula with the minimum loss.

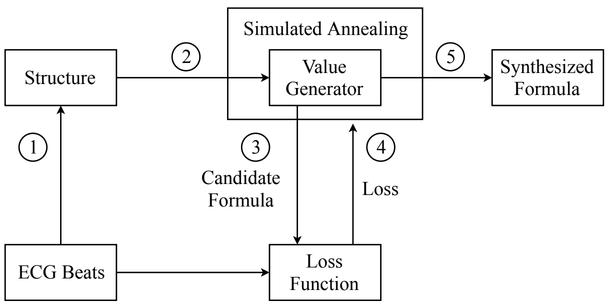
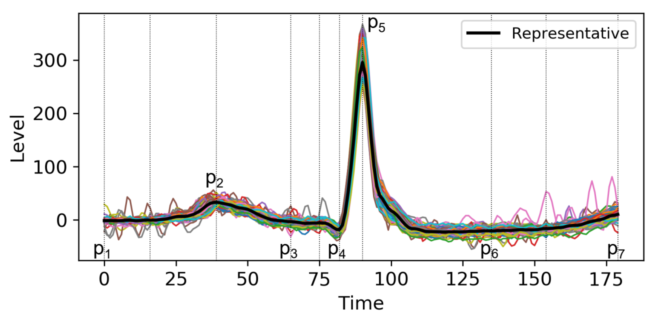


Fig. 1. Overview of the temporal logic synthesis method.

C. Automatic Structure Generation

As shown in Figure 2, the beat signals exhibit the patterns linearly along the time by its increase and decrease. Therefore, we reasonably limit the structures to the form "$\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$" where $\psi_i \in \{F[k_{i-1}+1, k_i]p_i, G[k_{i-1}+1, k_i]p_i\}$. Here p_i represents the local maximum and minimum points in the signals. The temporal operator $F[k_{i-1}+1, k_i]p_i$ represents the time to reach those points, while $G[k_{i-1}+1, k_i]p_i$ characterizes the signal as staying within the levels for a period of time. The time bound of $F[k_{i-1}+1, k_i]$ starts from $k_{i-1}+1$ where the event happens right after the previous. We will explain the algorithms of structure generation and use the example in Figure 2 to illustrate.



Raw signals (colored lines) are averaged to the representative (black line). Vertical dash lines show the segments in the composition algorithm with atomic propositions corresponding to the generated structure.

Fig. 2. Data example for automatic structure composition.

a) Segmentation: Given a time series trace $x(t)$, and its changing rate $\frac{dx}{dt}(t)$, we split it into a list of segments $\widehat{\mathcal{T}}$ by scanning through the rate of change, splitting the trace at the points where the sign changes among $\{0, +1, -1\}$. When there is noise in the data, unnecessary segments may appear and need to be removed. We then merges these noisy segments with the significant ones. For each of the segment in $\widehat{\mathcal{T}}$, if the change in the value of the trace variable throughout the time span is below a threshold (say 1%) of $(\max(x) - \min(x))$, it is merged into the segment before it. The segment list obtained after merging the noisy ones is denoted as $\mathcal{T}$. In Figure 2 the beat is splitted into the following segments (in the format of $(t_{begin}, t_{end}, sign)$ tuples) by the vertical dash lines: $(0, 15, 0)$, $(15, 40, 1)$, $(40, 67, -1)$, $(67, 75, 0)$, $(75, 81, -1)$, $(81, 88, 1)$, $(88, 134, -1)$, $(134, 155, 0)$, $(155, 180, 1)$.

b) Segments to Structure: We then translate the segment list $\mathcal{T}$ into the MITL structure S using operators $\{F, G, \wedge\}$ and atomic propositions p_τ in the form of $\ell_i \leq x \leq u_i$. Intuitively, if $sign_i$ of a segment $\tau_i \in \mathcal{T}$ is -1 or $+1$, the segment is translated to $F[k_{i-1}+1, k_i]p_i$ to signify that at the end of this segment, the value reaches p_i, representing a local maximum (or minimum) controlled by the constraint added to C. If the sign 0, the segment is translated to $G[k_{i-1}+1, k_i]p_i$ meaning the value remains in the range p_i during this time.

As an example, the segments from Figure 3 is translated into the following formula structure:

$$\begin{aligned} G[k_1, k_2]p_1 \wedge F[k_1+1, k_2]p_2 \wedge F[k_2+1, k_3]p_3 \\ \wedge G[k_3+1, k_4]p_3 \wedge F[k_4+1, k_5]p_4 \\ \wedge F[k_5+1, k_6]p_5 \wedge F[k_6+1, k_7]p_6 \\ \wedge G[k_7+1, k_8]p_6 \wedge F[k_8+1, k_9]p_7 \end{aligned}$$

where

$$p_i : \ell_i \leq x \leq u_i.$$

with constraints

$$u_1 < \ell_2, u_3 < \ell_2, u_4 < \ell_3, u_4 < \ell_5, u_6 < \ell_5, u_6 < \ell_7$$

Eq. 1. Formula structure from normal beat data.

This in essence is the formula (with values of the variables for each patient instantiated by the process described below) checked by our method for distinguishing normal from abnormal beats.

D. Parameter Synthesis

Given a generated structure $\psi = \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$ with the set of time bound variables $\mathrm{Var}_T = \{(k_1, k_1'), \ldots, (k_n, k_n')\}$, and the set of atomic propositions $\mathrm{Var}_{AP} = \{p_1, \ldots, p_n\}$, we automatically optimize the values associated with each member of Var_T and Var_{AP} such that the concretized formula gives the minimum loss.

A simulated annealing based procedure shown in Algorithm 1 is used to estimate the parameters.

1) **Value Generator:** We generate values for the propositional variables using the constraints specified in the propositional variables and the template constraints. Though the constraint satisfaction problem is NP-complete in general, the

Algorithm 1: optimizeParameter

Input : Template ψ
Output: Synthesized property ψ_{syn}
while ***Simulated Annealing*** *decides to continue* **do**
    $\widehat{\psi} \leftarrow$ **Value Generator**(ψ);
    Compute $Loss_{\widehat{\psi}} \leftarrow$ **Loss Function**($\widehat{\psi}, \mathcal{B}_{\widehat{\psi}}$);
    Simulated Annealing $\leftarrow Loss_{\widehat{\psi}}$;
    Update Var_T and Var_{AP} ;
return $\psi_{syn} \leftarrow \widehat{\psi}$ with minimum loss;

constraints in our framework are simple inequalities, allowing us to solve it using a tree-based approach. The value intervals of a variable are parsed as a tree structure where the values of the child nodes are larger than the parent nodes.

2) **MITL Checker:** After the process described above, the MITL formula Eq. 1 will have its variables instantiated with actual values for an individual patient. According to their definition, the checking of temporal operators $F[k, k']p$ and $G[k, k']p$ is derived as

$$F[k, k']p = \bigvee_{t=k}^{k'} p;$$
$$Gk, k']p = \bigwedge_{t=k}^{k'} p.$$

As we constrain the formula in the form of "$\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$" where $\psi_i \in \{F[k_{i-1}+1, k_i]p_i, G[k_{i-1}+1, k_i]p_i\}$, the computation cost for checking is linear to the length of signal, and the main operation is the comparison ($\ell \leq x \leq u$).

3) **Loss Function:** The loss is used to guide simulated annealing. It is consists of three components each for the temporal variables, atomic propositions and how well the formula fits the data.

Intuitively, a temporal operator $F[k, k']$ with a *smaller* time range $k' - k$ a stronger assertion because it is faster to reach a state that satisfies it. On the other hand, a temporal operator $G[k, k']$ with a *larger* range is stronger because the state is maintained for longer time. Therefore, the loss component L_T of the temporal variables is given by:

$$L_T = \prod_{(k_i, k'_i) \in \mathrm{Var}_T} \left(k'_i - k_i\right)^{sgn(k_i)}$$

$$sgn(k_i) = \begin{cases} -1, & \text{if temporal operator of } k_i \text{ is } G \\ 1, & \text{if temporal operator of } k_i \text{ is } F \end{cases}$$

where each pair (k_i, k'_i) are the time bounds of a temporal operator in the temporal parameter set Var_T.

For each atomic proposition, we consider both the magnitude of the value range, and how precisely it describes the behaviour of the trajectories.

For each atomic proposition "$p_i : \ell_i \leq x \leq u_i$" in the atomic proposition set Var_{AP}, we define the *tightness* as $(u_i - \ell_i)/(max_i - min_i)$, the range normalized to the maximum value range of the variable in all trajectories. The idea here is to keep the value range as small as possible.

Besides tightness, we also want a measure of how well the atomic propositions in $\widehat{\psi}$ fit the trajectories based on the constraints. Essentially, for each constraint of the form $u_j < \ell_k$ associated with the atomic propositions p_j and p_k, the estimated levels of p_j is expected to be lower than p_k. This information is also used to optimize $\widehat{\psi}$. To this end, we compute the mean value of p_i as $(\frac{\ell_i + u_i}{2})$. The weight w_i associated with each p_i is evaluated as follows. We first initialize the set of weights W_{AP} for all the atomic propositions in $\widehat{\psi}$ to 0. Then for each constraint $u_j < \ell_k$, we decrease w_j by 1 and increase w_k by 1. The fitness of an atomic proposition is therefore $\left(\frac{\ell_i + u_i}{2}\right)^{w_i}$. Intuitively, the level of p_j tends to be in the lower range of the value space while p_k to be in higher range.

Combining these two factors, we define the loss function component due to the propositional variables as

$$L_{AP} = \prod_{p_i \in \mathrm{Var}_{AP}} \left(\Big(\frac{u_i - \ell_i}{max_i - min_i}\Big) \Big(\frac{\ell_i + u_i}{2}\Big)^{w_i} \right).$$

Finally, in each iteration of the simulated annealing procedure, if the rate of satisfaction θ is larger than a pre-defined threshold h (in our case it was set to 0.95), we apply the loss function and continue with the iterations according to the search procedure. Otherwise, the loss is set as ∞ and the current combination of parameters is rejected. The search then continues with another set of parameters. Thus,

$$Loss_{\widehat{\psi}} = \begin{cases} L_{AP} \cdot L_T, & \theta > h \\ \infty, & \text{otherwise} \end{cases}.$$

III. Results and Discussion

A. Dataset and Preprocessing

In our evaluation, we use the Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) arrhythmia database [2], [3], which is a widely used benchmark for ECG analysis. This database has 48 ECG records of about 30 min each from 47 different patients.

ECG signals usually have noise from equipment, imperfection of operation, patient motion and so on. One of the primary noise sources is *baseline wander*, where the baseline of the x-axis shifts. There are many algorithms in removing the baseline wander [4]. In this study, we adopt the method in [5] by using two consecutive median filters to calculate the baseline of the signal, and subtracting it from the original signal. There have been much research on ECG segmentation to split the continuous records into individual beats [6], [7]. As this is orthogonal to our work, we simply use the given labeling locations of R peaks in the dataset for our purpose.

B. Classification Results

We apply our method to a few patients in the MIT-BIH dataset. For each beat type, the records are selected that they have sufficient amount abnormal signals. For each patient, we randomly choose half of the normal beats as training data and the other half as test data and all abnormal beats are used for testing. Table I shows the classification sensitivity of our methods. The dataset index refers to the record number and pair of beat types to classify. For example, 106-N,V refers to the N beats and V beats in record 106. The data size is the number of beats in the corresponding class for testing. For example, record 106 contains 753 N beats and 520 V beats. We used the CNN method from Li et al. as a comparison. A small portion of the patient's ECG record for is used for training, and a larger portion for testing [8]. From the table, our method outperforms the CNN in most patients.

TABLE I
EVALUATION OF PATIENT INDIVIDUAL CLASSIFICATION USING TEMPORAL LOGIC SYNTHESIS.

Dataset	**Data Size**	**Se**		**Acc**
		This	CNN [8]	
106-N,V	$753, 520$	94.23%	-	91.52%
208-N,V	$793, 992$	99.40%	94.36%	89.08%
213-N,V	$1281, 220$	100.00%	89.23%	91.74%
221-N,V	$1048, 396$	99.75%	99.37%	99.24%
207-N,S	$712, 106$	33.96%	0%	81.74%
209-N,S	$1297, 383$	28.20%	85.44%	76.90%
222-N,S	$1028, 420$	15.95%	0%	68.02%
208-N,F	$793, 372$	82.53%	74.34%	78.20%
213-N,F	$1281, 362$	57.73%	81.48%	83.14%

C. Computation Resource Comparison

Here we compare the resources used by our MITL checking method and the CNN [8] in terms of memory and computation as Table II.

1) Memory To store the MITL the formula "$\psi = \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$", each ψ_i consists 1 operator and 4 parameters stored as 4-byte integers and floats. In all our experiments, n is up to 10, so the required memory is 200 bytes. However in CNN, the number of weights to be stored is 13,728, which requires 54,912 bytes.
2) Computation Our MITL formula reasons the signal linearly. Therefore, in the worst case, for each data point, we perform 2 comparison for left and right bound and 2 boolean operations ($\wedge$ or $\vee$). Using the same input signal length of 196, we perform 392 comparisons and 392 boolean operations. In CNN, the main computation is the matrix multiplication and the comparison for ReLu activation. The network structure in [8] requires 15,047,168 multiplications, 15,037,563 additions, and 7,808 comparisons without further optimization.

As the preprocessing (including segmentation and noise removal) can be applied prior to both methods, the cost is excluded from the comparison. The computation cost comparison is summarized in Table II. MITL checking requires only 0.3% of memory and 5% of comparison operations, compared with CNN. It also saves millions of multiplications and additions. Therefore, it is much more suitable for on-device classification.

TABLE II
COMPARISON OF MITL CHECKING WITH CNN INFERENCE

	Memory (Bytes)	Bool	Compare	Multiply	Add
MITL	200	392	392	0	0
CNN	54,912	0	7,808	15,047,168	15,037,563

Cost of signal segmentation and noise removal are excluded in both methods.

IV. CONCLUSION

In this paper, we have proposed the use of temporal logic synthesis to detect the onset of abnormal heart beats from personalized ECG signals. Given a set of normal ECG beats, we first generate the formula structure with the time and atomic proposition bounds as parameters, and then optimize the parameters to generate the concrete formula. Used as a classifier, normal beats will satisfy the final concrete formula while abnormal ones will not.

We have demonstrated that our method outperforms the CNN in both accuracy and resource usage. Specifically, our method uses only 0.3% of memory and 5% of comparison operations, compared with CNN. It also avoids millions of multiplications and additions. Therefore, it is much more suitable for implementation on wearable devices. Our low cost method complements other techniques for detecting abnormal heart rates and/or rhythm, all of which can be combined into a robust and energy efficient scheme for detecting the onset of heart irregularities using resource-constrained wearable devices, potentially saving many lives.

REFERENCES

[1] A. Dokhanchi, B. Hoxha, and G. Fainekos, "Metric interval temporal logic specification elicitation and debugging," in *2015 ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE)*. IEEE, 2015, pp. 70–79.

[2] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[3] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[4] F. P. Romero, L. V. Romaguera, C. R. Vázquez-Seisdedos, M. G. F. Costa, J. E. Neto *et al.*, "Baseline wander removal methods for ecg signals: A comparative study," *arXiv preprint arXiv:1807.11359*, 2018.

[5] V. Mondéjar-Guerra, J. Novo, J. Rouco, M. Penedo, and M. Ortega, "Heartbeat classification fusing temporal and morphological information of ecgs via ensemble of classifiers," *Biomedical Signal Processing and Control*, vol. 47, pp. 41–48, 2019.

[6] A. Zifan, S. Saberi, M. H. Moradi, and F. Towhidkhah, "Automated ecg segmentation using piecewise derivative dynamic time warping," *International Journal of Biological and Medical Sciences*, vol. 1, no. 3, 2006.

[7] J. Aspuru, A. Ochoa-Brust, R. A. Félix, W. Mata-López, L. J. Mena, R. Ostos, and R. Martínez-Peláez, "Segmentation of the ecg signal by means of a linear regression algorithm," *Sensors*, vol. 19, no. 4, p. 775, 2019.

[8] Y. Li, Y. Pang, J. Wang, and X. Li, "Patient-specific ecg classification by deeper cnn from generic to dedicated," *Neurocomputing*, vol. 314, pp. 336–346, 2018.