Runtime Verification for Clinically Interpretable Arrhythmia Classification

Alex Baird
University of Auckland
Auckland, New Zealand
alex.baird@auckland.ac.nz

Srinivas Pinisetty
I.I.T. Bhubaneswar
Jatni, Odisha, India
spinisetty@iitbbs.ac.in

Nathan Allen
University of Auckland
Auckland, New Zealand
nathan.allen@auckland.ac.nz

Nitish Patel
University of Auckland
Auckland, New Zealand
nd.patel@auckland.ac.nz

Partha Roop
University of Auckland
Auckland, New Zealand
p.roop@auckland.ac.nz

***Abstract*—Automatic detection of cardiac arrhythmia is an important tool in the fight against cardiovascular diseases and their associated human impacts. Such detection needs to be both accurate and timely, in order to allow for interventions to be administered within short time frames. Traditionally, such approaches have used black box implementations which are not explainable and hence have limited use in terms of clinical interpretability. Additionally, these implementations may either require additional training between patients, or have processing times which make them unsuitable for real-time classification.**

To address this, we develop a set of formal Timed Automaton-based policies that capture three common arrhythmia, Premature Ventricular Contraction, Ventricular Tachycardia, and Atrial Fibrilation, in terms of Electrocardiogram (ECG) features. We synthesise Runtime Verification monitors for each of these policies, and run them alongside existing clinical ECG databases to evaluate their efficacy. This approach shows comparable results to existing black box work with accuracies ranging from 90 % to 96 % while still being both explainable and clinically interpretable.

***Index Terms*—Arrhythmia Classification, Runtime Verification, Timed Automata, Explainability, ECGs**

I. Introduction

Cardiovascular disease is the leading cause of death worldwide [1], many of which are sudden cardiac death and occur outside of hospitals [2]. Unfortunately, for those patients under 35, this is usually caused by an arrhythmia in an otherwise structurally normal heart and many are asymptomatic [3]. This shows a clear need for early detection and intervention.

Traditionally, clinicians use Electrocardiograms (ECGs) to monitor the electrical activity of the heart from the body surface, which are interpreted to assess the rhythm and timing of a patient's heart. An ECG wave, shown in Figure 1, can be separated into five typical components, P, Q, R, S, and T waves, also known as ECGs events. To support clinicians in interpreting ECGs, extensive computer models and tools have been developed.

These existing models and analysis techniques are increasingly incorporating machine learning techniques [4], [5], [6], [7]. These approaches are trained on large databases of "labelled" ECG recordings. The labels, added by trained clinicians, indicate which, if any, arrhythmias are present in each recorded trace. Once trained, the models classify previously unseen ECGs based on the patterns learned during the training phase. These approaches often focus on a single arrhythmia rather than multiple, have limited accuracy due to the variation within training data, provide no guarantees of correctness, and are "black boxes." Black box systems are those where the internal process is unknown and so can only be considered in terms of input and output. In the context of ECG interpretation this means there is no explanation as to why any given ECG was classified how it was. This limits clinical interpretation which is useful for informing a diagnosis and building confidence in an assessment. There is therefore an opportunity for the development of robust and clinically interpretable ECG analysis techniques which are instead "white box" in nature.

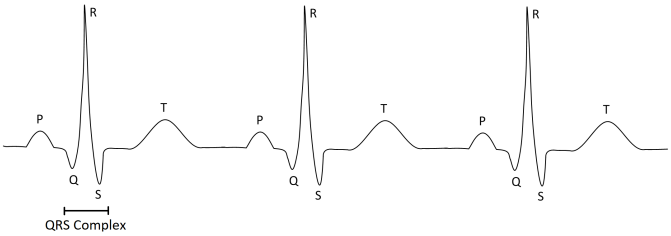


Fig. 1: Three heart beats of an ECG with labelled P, Q, R, S, and T waves. A QRS complex consists of a Q, R, and S wave.

Formal methods [8] are mathematically robust techniques to describe, develop, and verify systems, typically applied when performance and safety need to be guaranteed. Recently, they have been used in medical applications for pacemakers [9], [10] and insulin pumps [11]. Expressing a range of arrhythmia indicators present in ECGs using a formal expression would produce a system with soundness guarantees.

Runtime Verification (RV) [12], [13] is the monitoring of policies that a system should satisfy during execution. If a

policy is violated, the monitor will raise an alarm. In this context, as illustrated in Figure 2, the system is a human heart which produces the electrical signals captured by an ECG. Policies can then be defined to capture the features of various arrhythmia present in an ECG. RV enables real-time verification of these policies on a portable device, raising an alarm if an arrhythmia is detected. Importantly, such systems have known internal processes and are therefore classified as white box systems — meaning that they can be useful for clinical interpretation.

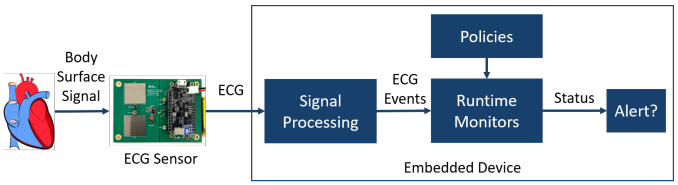


Fig. 2: Block diagram of Runtime Verification system for ECG classification

We, for the first time, present arrhythmia specifications formally for Premature Ventricular Contractions (PVCs), Ventricular Tachycardia (VT), and Atrial Fibrilation (AF) as Timed Automata (TAs) [14], allowing us to apply rigorous formal approaches such as RV. TA specifications are based on the key indicators of arrhythmia described by Thaler in [15]. From the TA arrhythmia specifications, RV monitors are automatically synthesised using techniques proposed in [16], to provide explainable ECG monitoring that can classify ECG traces in real-time.

These contributions are presented as follows. In Section II we will first present background information around TAs and RV. Subsequently, in Section III, we describe the process for modelling each policy used for the classification of arrhythmia in ECGs, along with an overview of how the overall system would look in Section IV. Next, we compare this approach to existing work in Section V using labelled ECG traces that were extracted from reputable databases. Section VI contains a discussion of the related work and Section VII contains concluding remarks.

II. Preliminaries and Background

Runtime Verification (RV) is based on the monitoring of input streams σ, and we begin by formalising these streams as words over an alphabet Σ. We then expand the setup to include timed words. A finite word over Σ is a finite sequence $\sigma = a_1 \cdot a_2 \cdots a_n$ of elements of Σ. $|\sigma|$ signifies the length of σ, and ϵ denotes an empty word over Σ. Any subset $\mathcal{L} \subseteq \Sigma^*$ is an language over alphabet Σ, where Σ^* represents the set of all words over Σ.

The concatenation of two words σ and σ' is indicated as $\sigma \cdot \sigma'$. A word σ' is a prefix of a word σ, denoted as $\sigma' \preccurlyeq \sigma$, if there exists a word σ'' such that $\sigma = \sigma' \cdot \sigma''$, and if additionally $\sigma' \neq \sigma$ then $\sigma' \prec \sigma$; conversely σ is called an *extension* of σ'.

The *set of prefixes* of σ is denoted as $\mathrm{pref}(\sigma)$ and subsequently, $\mathrm{pref}(\mathcal{L}) \stackrel{\text{def}}{=} \bigcup_{\sigma \in \mathcal{L}} \mathrm{pref}(\sigma)$ denotes the set of prefixes of words in $\mathcal{L}$. A language $\mathcal{L}$ is *extension-closed* if $\mathcal{L} \cdot A^* = \mathcal{L}$ and *prefix-closed* if $\mathrm{pref}(\mathcal{L}) = \mathcal{L}$.

A. Timed words and timed languages

When considering a timed framework, it's also crucial to examine the instants in time when actions occur. Let Σ be a finite collection of actions, and $\mathbb{R}_{\geq 0}$ be the set of non-negative real numbers. The pair (t, a) is an event, where $\mathrm{date}((t,a)) \stackrel{\text{def}}{=} t \in \mathbb{R}_{\geq 0}$ is the absolute time of the event, and $\mathrm{act}((t,a)) \stackrel{\text{def}}{=} a \in \Sigma$ is the action.

A finite sequence of events $\sigma = (t_1, a_1) \cdot (t_2, a_2) \cdots (t_n, a_n)$ is a timed word over the alphabet Σ, with $(t_i)_{i \in [1,n]}$ being a non-decreasing sequence in $\mathbb{R}_{\geq 0}$. $\mathrm{start}(\sigma) \stackrel{\text{def}}{=} t_1$ signifies σ's beginning date, and $\mathrm{end}(\sigma) \stackrel{\text{def}}{=} t_n$ denotes σ's ending date. For ϵ (empty timed word), the beginning and ending instants in time are both null.

Given an alphabet Σ, $\mathrm{tw}(\Sigma)$ represents the set of timed words over Σ, and any set $\mathcal{L} \subseteq \mathrm{tw}(\Sigma)$ is a timed language. Although the alphabet in the timed environment $(\mathbb{R}_{\geq 0} \times \Sigma)$ is infinite, notations in the untimed setting relating to length, prefix, and so on extend to timed words.

The *untimed projection* (i.e. ignoring time) of σ is $\Pi_\Sigma(\sigma) \stackrel{\text{def}}{=} a_1 \cdot a_2 \cdots a_n$ in Σ^*. When concatenating two timed words, the instants in time should be non-decreasing in the resulting timed word. This is ensured if the ending date of the first timed word is less than the starting date of the second timed word. Formally, consider $\sigma = (t_1, a_1) \cdots (t_n, a_n)$ and $\sigma' = (t'_1, a'_1) \cdots (t'_m, a'_m)$ to be two timed words with $\mathrm{end}(\sigma) \leq \mathrm{start}(\sigma')$. Their concatenation is

$$\sigma \cdot \sigma' \stackrel{\text{def}}{=} (t_1, a_1) \cdots (t_n, a_n) \cdot (t'_1, a'_1) \cdots (t'_m, a'_m)$$

By convention $\sigma \cdot \epsilon \stackrel{\text{def}}{=} \epsilon \cdot \sigma \stackrel{\text{def}}{=} \sigma$. Concatenation is undefined otherwise.

B. Properties of Timed Automaton

A finite automaton that is extended with a finite set of real-valued clocks $X = \{x_1, \ldots, x_k\}$ is called a Timed Automaton (TA) [17]. A function χ from X to $\mathbb{R}_{\geq 0}$, which is an element of $\mathbb{R}^X_{\geq 0}$ is a *clock valuation* for X. We have $\chi + \delta$ representing the valuation assigning $\chi(x) + \delta$ to each clock x of X, given $\chi \in \mathbb{R}^X_{\geq 0}$ and $\delta \in \mathbb{R}_{\geq 0}$. For a given set of clocks $X' \subseteq X$, the clock valuation χ where all clocks in X' are assigned to 0 is represented as $\chi[X' \leftarrow 0]$. The set of *guards* are denoted as $\mathcal{G}(X)$, with clock constraints defined as conjunctions of constraints of the form $x \bowtie c$ with $x \in X$, $c \in \mathbb{N}$ and $\bowtie \in \{<, \leq, =, \geq, >\}$. When g holds as per χ for a given $g \in \mathcal{G}(X)$ and $\chi \in \mathbb{R}^X_{\geq 0}$, it is denoted as $\chi \models g$.

1) Timed Automaton syntax and semantics: The properties to be verified are formalised as TAs in this study, and RV monitors are synthesised from them.

Definition II-B.1 (Timed Automaton): A Timed Automaton (TA) $\mathcal{A} = (L, l_0, X, \Sigma, \Delta, F)$ is a tuple, where L is a finite set of *locations*, $l_0 \in L$ is the *initial location*, X is a finite set of *clocks*, Σ is a finite set of *actions*, and $\Delta \subseteq L \times \mathcal{G}(X) \times \Sigma \times$

$2^X \times L$ is the *transition relation*. $F \subseteq L$ is a set of *accepting locations*, and locations in $L \setminus F$ are non-accepting.[1]

A TA has an infinite number of states since the set of possible values for a clock is infinite. The semantics of a TA is described as a transition system wherein every state is a tuple comprised of the present location and clock values. The following is a definition of its semantics.

Definition II-B.2 (Semantics of a Timed Automaton): The semantics of a Timed Automaton (TA) $[\![\mathcal{A}]\!] = (Q, q_0, \Gamma, \rightarrow, Q^F)$ is a *timed transition system* where $Q = L \times \mathbb{R}^X_{\geq 0}$ is the (infinite) set of *states*, $q_0 = (l_0, \chi_0)$ is the *initial state* where χ_0 is the valuation mapping every clock in X to 0, $\Gamma = \mathbb{R}_{\geq 0} \times \Sigma$ is the set of transition *labels*, that are pairs composed of a delay and an action, and $Q^F = F \times \mathbb{R}^X_{\geq 0}$ is the set of *accepting states*. $\rightarrow \subseteq Q \times \Gamma \times Q$ is the *transition relation*, corresponding to the set of transitions of the form $(l, \chi) \xrightarrow{(\delta, a)} (l', \chi')$ with $\chi' = (\chi + \delta)[Y \leftarrow 0]$ whenever $\exists (l, g, a, Y, l') \in \Delta$ such that $\chi + \delta \models g$ for $\delta \in \mathbb{R}_{\geq 0}$.

A *run* of $\mathcal{A}$ from a state $q \in Q$ to a state $q' \in Q$ over a timed trace $\sigma = (t_1, a_1) \cdot (t_2, a_2) \cdots (t_n, a_n)$ is a sequence of transitions $q_0 \xrightarrow{(\delta_1, a_1)} q_1 \cdots q_{n-1} \xrightarrow{(\delta_n, a_n)} q_n$, where $q_0 = q$, $q_n = q'$, $\delta_1 = t_1$, and $\forall i \in [2, n] : \delta_i = t_i - t_{i-1}$. If there exists a run from q to q' over σ, it is denoted by $q \xrightarrow{\sigma} q'$.

$\mathcal{L}(\mathcal{A}, q, K)$ denotes a *language* of $\mathcal{A}$, starting in q and ending in K (for $q \in Q$ and $K \subseteq L$), defined by

$$\mathcal{L}(\mathcal{A}, q, K) = \{w \mid \exists q' \in K \times \mathbb{R}^X_{\geq 0} : q \xrightarrow{w} q'\}$$

The overall language of $\mathcal{A}$, denoted as $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}, q_0, F)$ is the language of $\mathcal{A}$ starting from q_0 (initial state) and ending in a state F.

An example TA which captures a wide QRS complex, a feature of PVC, is shown in Figure 3 as policy φ_1. Here, we have an input alphabet $\Sigma = \{QRS\_START, QRS\_END\}$, and the policy checks that a QRS_START is followed by a QRS_END within 120 ms. The set of locations shown is $L = \{l0, l1, l2, l3\}$, the initial location $l_0 = l0$, and the accepting locations are $F = \{l0, l1\}$ which leaves the non-accepting locations as $L \setminus F = \{l2, l3\}$. There exists only one clock, meaning that the set of clocks is $X = \{t\}$. The single clock is reset on transitions from $l0$ to $l1$ and from $l3$ to $l4$. An example guard is $t > 120$, measured in milliseconds, on the transition between $l1$ and $l2$. The automaton transitions from $l1$ to a non-accepting location $l2$ when the QRS_END event is observed and the value of clock $t > 120$.

C. Runtime Verification Monitor

Let us consider the definition of an RV monitor for any given specified timed policy φ in this section. The monitor for φ takes the current observation σ as input and emits a verdict from the set $\mathcal{D} = \{\mathsf{c\_true}, \mathsf{c\_false}\}$, where c_true (currently true) indicates that the current observed behaviour satisfies φ, and c_false (currently false) indicates that current observed behaviour does not satisfy φ.

[1]In addition to this definition, to simplify our TA models we use the concept of internal variables for computation within the TA, e.g. to keep track of the number of events received. Transitions can have guards and updates (e.g. increment or reset) over the internal variables.

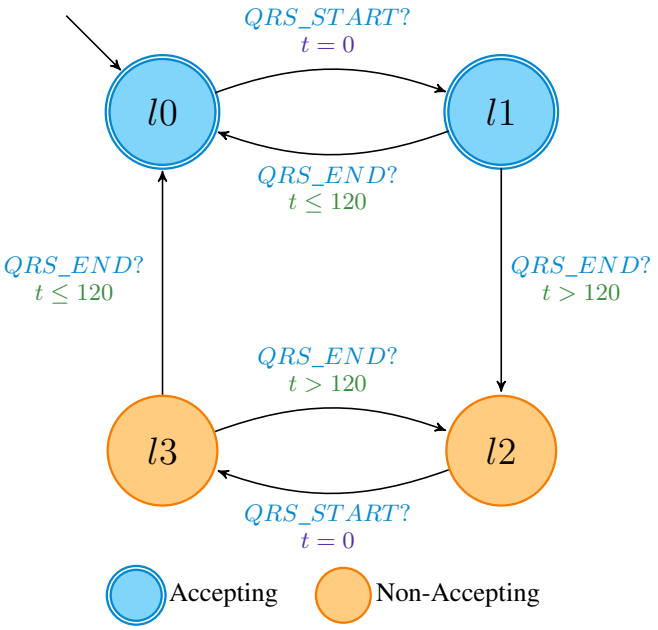


Fig. 3: Policy φ_1 which captures the feature of a wide QRS complex

Definition II-C.1 (Runtime Verification monitor): Consider the policy $\varphi \subseteq \mathrm{tw}(\Sigma)$ which defines the policy to be monitored and which is defined as a TA $\mathcal{A}_\varphi$. Function $M_\varphi : \mathrm{tw}(\Sigma) \rightarrow \mathcal{D}$ is a verification monitor for φ and is defined as follows, with $\sigma \in \mathrm{tw}(\Sigma)$ denoting the current observation (a finite timed word over the alphabet Σ):

$$M_\varphi(\sigma) = \begin{cases} \mathsf{c\_true} & \text{if } \sigma \in \varphi \\ \mathsf{c\_false} & \text{if } \sigma \notin \varphi \end{cases}$$

Using the previously described policy φ_1, shown in Figure 3, we can show the outputs of an RV monitor in response to a given input trace. Take, for example, the input trace σ = (50, START) · (150, QRS_END) · (350, QRS_START) · (480, QRS_END) · (680, QRS_START) · (770, QRS_END). Table I shows the inputs at each time instant, and the monitor's response, to the input σ. The trace consists of three heart beats. The second beat is a wide beat, as the time between QRS_START and QRS_END is greater than 120 ms, and so the monitor verdict is c_false for that beat.

TABLE I: Example event Trace for Policy φ_1

Loc.	t	Event Pairs	Trans.	Clock	$M_{\varphi_1}(\sigma)$
l0	0	-	l0 → l0	-	c_true
l0	50	50, QRS_START	l0 → l1	t = 0	c_true
l1	100	150, QRS_END	l1 → l0	-	c_true
l0	300	350, QRS_START	l0 → l1	t = 0	c_true
l1	130	480, QRS_END	l1 → l2	-	c_false
l2	330	680, QRS_START	l2 → l3	t = 0	c_false
l3	90	770, QRS_END	l3 → l0	-	c_true

In our scenario, we have multiple policies to be monitored for each single arrhythmia, and so the question arises: how do

we combine them? One possibility to monitor all policies is the monolithic approach where we first combine all the policies by computing the product of all TAs specifying the policies. The monolithic approach has obvious limitations such as state-space explosion, and prevents the localisation of knowing which specific policy lead to a violation.

Therefore, in this work we consider obtaining an individual monitor for each policy and using a "merge function" to combine the verdicts provided by all monitors.

Definition II-C.2 (Merge Function: Δ):

Function $\Delta : (\mathcal{D}_1 \times \mathcal{D}_2 \cdots \times \mathcal{D}_n) \rightarrow \mathcal{D}$ is defined as the disjunction of events, formally represented as:
Let $\forall i \in [1, n], d_i \in \mathcal{D}$.

$$\Delta(d_1, \cdots, d_n) \quad = \begin{cases} \mathsf{c\_true} & \text{if } \exists i \in [1, n], d_i = \mathsf{c\_true} \\ \mathsf{c\_false} & \text{if } \forall i \in [1, n], d_i = \mathsf{c\_false} \end{cases}$$

The merge function Δ returns c_true while any one verdict is c_true and returns c_false if all verdicts are c_false. Only when all monitors simultaneously emit verdict c_false do we classify the arrhythmia as present. This reflects the requirement that all features of arrhythmia are present simultaneously for the classification.

Considering the merge function and the need to simultaneously monitor multiple policies for arrhythmia, we define composition for RV monitors as follows.

Definition II-C.3 (Composition of Runtime Verification monitors): Given $\varphi_1, \varphi_2, \cdots, \varphi_n$, where $\forall i \in [1, n], \varphi_i \subseteq \text{tw}(\Sigma)$, let $M_{\varphi_1}, M_{\varphi_2}, \cdots, M_{\varphi_n}$ denote their corresponding RV monitors. The composition of monitors $M_{\varphi_1}, M_{\varphi_2}, \cdots, M_{\varphi_n}$ is denoted as $M_{\varphi_1}||M_{\varphi_2}||\cdots||M_{\varphi_n}$. It is expressed as a function $M_{\varphi_1}||M_{\varphi_2}||\cdots||M_{\varphi_n} : \text{tw}(\Sigma) \rightarrow \mathcal{D}$, defined as follows:

$$M_{\varphi_1}||M_{\varphi_2}||\cdots||M_{\varphi_n}(\sigma) = \Delta\Big(M_{\varphi_1}(\sigma),\ M_{\varphi_2}(\sigma), \cdots, M_{\varphi_n}(\sigma)\Big)$$

Using these definitions, we can now define monitors that each capture a separate arrhythmia feature, compose individual verdicts using the merge function, and produce a single verdict per arrhythmia. In the following section we define policies for a range of arrhythmia.

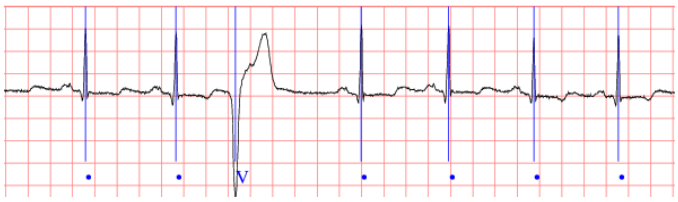
III. Capturing Arrhythmia as Timed Automata

The use of TAs [14] is well suited to capturing the timing relationship of ECG events formally. Additionally, because events are meant to arrive in a pre-determined order (P, Q, R, S, T), TAs are able to capture the absence of events, allowing for the basic morphology of an ECG to be captured.

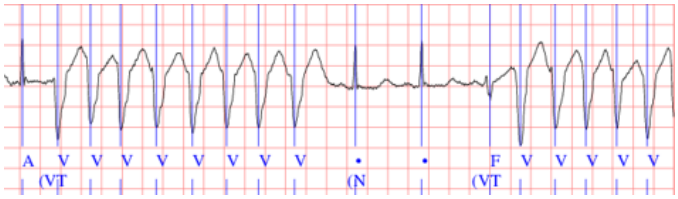
We use TAs to formalise policies which, when composed, capture the features of arrhythmia as defined in [15], [18]. Thaler's work in [15] presents the identifying features of arrhythmia in a manner enabling students to learn to read ECGs. We build TA-based policies from these arrhythmia features and go on to synthesise runtime monitors to classify ECG traces.

When a TA is in an accepting location (monitor output is c_true), the ECG is classified as not displaying the feature of arrhythmia captured by that policy. When in a non-accepting location (monitor output is c_false), the ECG is classified as displaying that feature of arrhythmia.

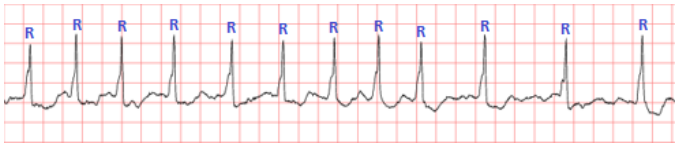
In this work we develop a formal specification for the following, where each policy is related to one or more arrhythmia: φ_1 and φ_2 for Premature Ventricular Contraction (PVC), φ_3 and φ_4 for Ventricular Tachycardia (VT), and φ_5 and φ_2 for Atrial Fibrilation (AF). For each of these arrhythmia, all features need to be present to classify the ECG as displaying the arrhythmia. Therefore, the overall specification considered for monitoring each arrhythmia is the composition of monitors, as previously described in Definition II-C.3. The specifications are defined using the event alphabet $\Sigma = \{R, P, QRS\_START, QRS\_END\}$. The R and P events are the peaks of their corresponding ECG waves. The QRS_START event is the first deviation from the base line for the Q wave, while the QRS_END event is the final deviation from the base line for the S wave.



(a) Premature Ventricular Contraction



(b) Ventricular Tachycardia



(c) Atrial Fibrilation

Fig. 4: Example ECG traces from the MIT-BIH database [19] showing various arrhythmia

A. Premature Ventricular Contraction

PVC, the most common arrhythmia, occurs when ventricular depolarisation takes an abnormal conduction pathway. This results in a wide QRS complex, which consists of the Q, R, and S waves, with an abnormal morphology as visualised in the trace in Figure 4a. A wide QRS complex is defined as one with duration greater than $120\,\text{ms}$ [15]. Additionally, it is common for the P wave to be absent during PVC. While isolated PVCs are common and rarely require treatment, they are a component of more serious arrhythmias. For example bigeminy and trigeminy, where PVCs occur commonly and in

patterns with sinus beats, and VT, which is discussed in the next section.

Policy φ_1, shown in Figure 3, was developed to detect a wide QRS complex. This policy considers the QRS_START and QRS_END events to determine the width of the QRS complex. After the QRS_START event, the clock t is reset and when the QRS_END event is received the automaton transitions to either the accepting location *l0* if the timer value was less than or equal to $120\,\mathrm{ms}$, or to the non-accepting location *l3* if the timer value was above $120\,\mathrm{ms}$. After each QRS complex the automaton will be in either an accepting or non-accepting location classifying that QRS complex as normal or wide respectively.

Policy φ_2, shown in Figure 5, captures the absence of P waves. This policy considers two input events, the P wave and the R wave, and checks that for every R wave there should be a P wave observed prior. If the P wave is not observed, the automaton transitions into the non-accepting location *l3*. If at any point the P waves return, the automaton transitions to the accepting location *l2*.

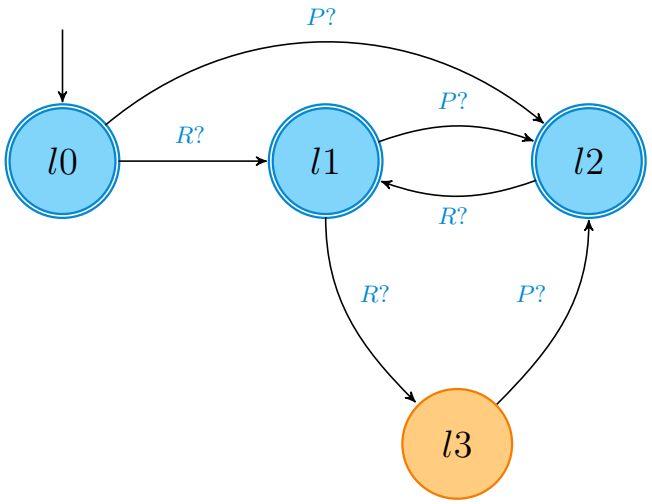


Fig. 5: Policy φ_2 captures the feature of P waves being missed before R waves

The composition of policies φ_1 and φ_2 is taken to detect the presence of a PVC beat.

B. Ventricular Tachycardia

VT is defined as three or more PVCs in a row, often with a rate between 120 and 200 beats per minute [15], as illustrated by the trace in Figure 4b. An increased risk of VT exists in individuals who have experienced a heart attack. In order to detect VT, two policies were created which each extend the previous φ_1 and φ_2 policies to capture the occurrence of three consecutive PVCs.

Policy φ_3, shown in Figure 6, captures the absence of P waves in three consecutive beats, extending φ_2 while using the same set of input events. The policy remains in accepting locations until three consecutive R waves have occurred without a P wave, upon which it transitions to location *l2*. The internal variable *i* is used to count the number of consecutive missing P waves. If a P wave is detected between any R wave the automaton transitions to the accepting location *l0* and on the next R wave the counter *i* is reset to zero. This could happen either from the non-accepting location *l2* or the accepting location *l1*.

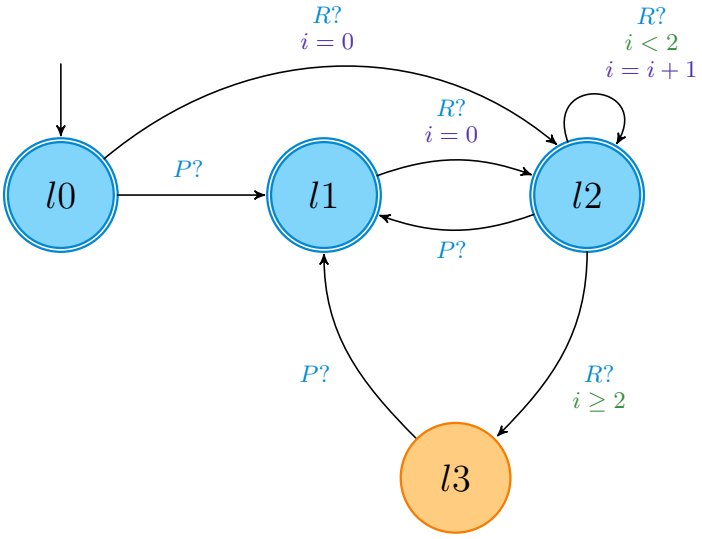


Fig. 6: Policy φ_3 captures the feature of three successive R waves missing P waves.

Policy φ_4, shown in Figure 7, captures three consecutive wide QRS complexes, extending φ_1 while using the same set of input events. The policy remains in accepting locations *l0* and *l1* until three consecutive beats with wide QRS complexes are detected. The internal variable *i* is used to count the number of consecutive wide QRS complexes. On each QRS_END event the timer and counter are considered. If the timer is less than $120\,\mathrm{ms}$ (a normal width) the transition to *l0* is taken and the counter *i* is reset to zero. If the timer is greater than or equal to $120\,\mathrm{ms}$ (a wide width) the counter is considered. If the counter has a value less than two there have not been three consecutive wide complexes so the transition to *l0* is taken and the counter is incremented. If the counter is equal or greater than two, the policy transitions to the non-accepting location *l2*, where it continues to be non-accepting until a QRS complex with width less than $120\,\mathrm{ms}$ is observed.

The composition of policies φ_3 and φ_4 is taken to detect the presence of VT.

C. Atrial Fibrilation

AF is caused by multiple re-entrant circuits, where the electrical impulse travels unpredictably in a circular loop resulting in no discernible P waves in the ECG. The Atrioventruclar (AV) node, responsible for conduction between the atria and ventricles, can only respond to some of these impulses and does so at variable intervals resulting in irregular R waves. These two features can both be seen in the ECG trace in Figure 4c. The previously introduced policy φ_2 (Figure 5)

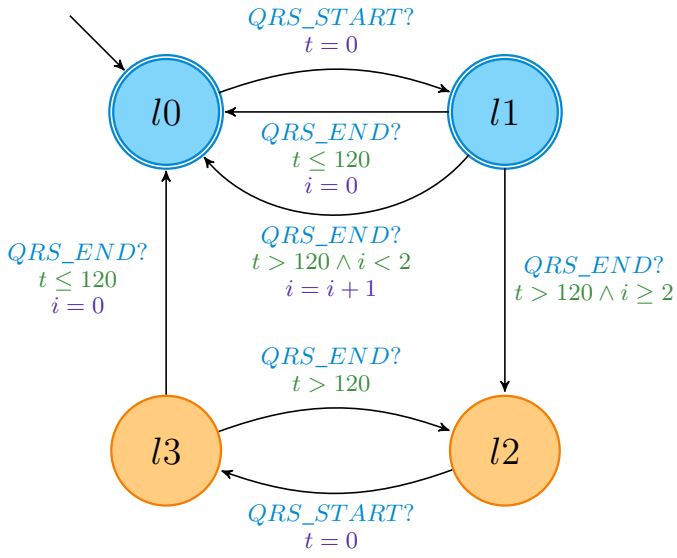


Fig. 7: Policy φ_4 captures the feature of three successive wide QRS complexes

is used to capture the first feature, while a further TA-based policy is created to capture the irregular R waves of AF.

Policy φ_5, shown in Figure 8, captures the irregularity in R waves, by using two clocks which capture the time since the two most recent R waves, i.e. t_{n-2} and t_{n-1}. The two clocks $t1$ and $t2$ are updated in a lock-step manner, meaning that in locations *l3* and *l4*, $t1$ and $t2$ correspond to t_{n-1} and t_{n-2} respectively, while in locations *l2* and *l5*, they correspond to t_{n-2} and t_{n-1} respectively. On each received R wave, the variance between the two preceding R-R intervals is calculated using the function δ shown in Equation 1, where $t_{n-2} < t_{n-1}$ (i.e. t_{n-2} captures the earlier event). If this difference is above some maximum value, δ_{max} in Figure 8, the automaton transitions into a non-accepting location *l4* or *l5* as the waves are said to be irregular. If this difference is, or returns to being, within the acceptable range then the automaton transitions to an accepting location *l2* or *l3*.

$$\delta(t_{n-2}, t_{n-1}) = \left|(t_{n-2} - t_{n-1}) - t_{n-1}\right| \tag{1}$$

The δ_{max} time is set to $55\,\text{ms}$, which was selected based on the interval between R waves compared across AF and sinus rhythm traces from the MIT-BIH database [19], [20]. Importantly, this interval performed acceptably across multiple databases as detailed later in Section V. This interval can also readily be changed, for example by a clinician for a specific patient, or to an interval based on a wider analysis across populations.

The composition of policies φ_2 and φ_5 is taken to detect the presence of AF.

IV. Real Time Arrhythmia Classification

Using the policies previously defined in Section III and the framework used in [16], [21] we are able to synthesise runtime monitors. These monitors become part of a RV-based framework which is continually checking to see that the set of properties are satisfied. In this context the system which is being verified is an ECG, capturing the electrical activity of the patient's heart in real time.

The overall RV-based arrhythmia classification system designed is illustrated in Figure 2. A patient would provide their ECG via a small portable ECG recorder, or wearable device, which is then passed into the embedded device which performs signal processing to produce timed events in near real time. These events are fed into each of the runtime monitors, synthesised from the arrhythmia specification policies φ_1 through φ_5, which each output determine whether the feature is present or not.

The outputs of these monitors are then combined, through the use of the merge function defined in Section II-C, to provide verdicts for each arrhythmia being monitored. It is important to remember that for the system to provide a verdict that the ECG contains a given arrhythmia requires that *all* features are present simultaneously. This results in a system which produces specific alerts based on the detection of either PVC, VT, or AF in an ECG in real-time.

One aspect which has not been discussed in detail is that of signal processing. This paper does not look at or compare various methods for extracting events from ECGs, and instead makes use of off the shelf toolboxes for the testing and validation of this approach, as discussed in the next section.

V. Results

To validate our arrhythmia specifications we used ECGs from the following clinical databases: MIT-BIH [19], MIT-BIH AF [22], CPSC2018 [23], and an AAEL sample subset [24]. These databases all contain clinician labelled ECG traces which allow us to measure the performance of our complete system as illustrated in Figure 9.

Signal processing, through filtering and wave detection, produces timed ECG events based on the selected clinical recording. We used the open source ECGdeli delineation toolbox for MATLAB [25]. In addition to the standard ECG filtering of a $120\,\text{Hz}$ low pass, $49\,\text{Hz}$ to $51\,\text{Hz}$ band stop, and a $0.3\,\text{Hz}$ high pass, we added wavelet de-noising for further noise reduction.

From this signal processing, timed ECG events are passed into each of the runtime monitors. The arrhythmia specifications were implemented as TA-based policies in UPPAAL [26] and synthesised into runtime monitors using the UPPAAL Python libraries and an associated Python framework presented in [16]. This provided the environment for evaluating the signal processing and arrhythmia specifications using clinical ECGs.

Performance was measured using accuracy, sensitivity, and specificity. Accuracy is the percentage of traces that were correctly classified over the entire dataset. Sensitivity (Equation 2) is relevant for screening, and is the percentage of traces correctly classified as having the arrhythmia from the group of traces with the arrhythmia, in other words the probability of a

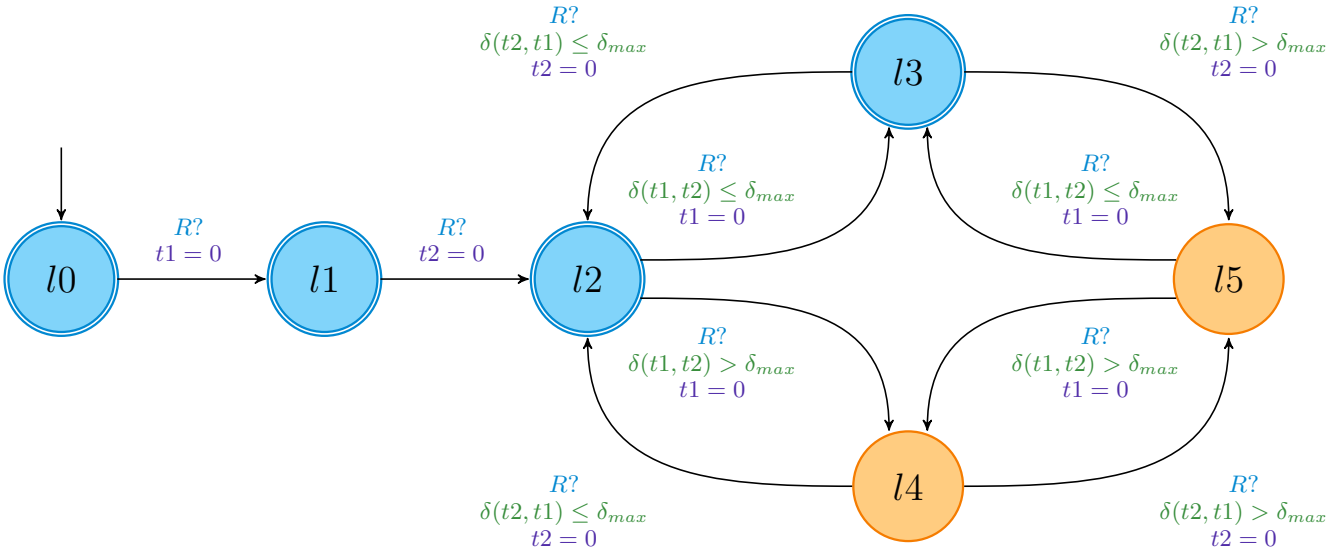


Fig. 8: Policy φ_5 captures the feature of a varying interval between R waves

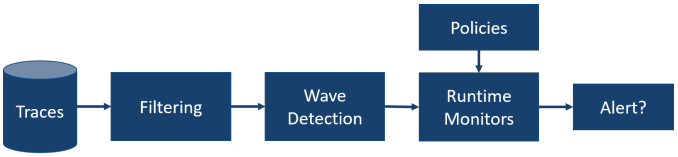


Fig. 9: Block diagram of system to evaluate arrhythmia specifications as Runtime Verification monitors

positive classification given that the trace has the arrhythmia. Specificity (Equation 3) is relevant for diagnosis, and is the probability that a trace without the arrhythmia is correctly classified as not having the arrhythmia.

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives} \quad (3)$$

A challenge of comparison with existing work was the availability and range of databases and performance metrics used. We endeavoured to evaluate our work using a range of databases including, where possible, those used by others for fairer comparison.

A. Premature Ventricular Contractions

The PVC detection was assessed using the MIT-BIH [19] and CPSC 2018 databases [23]. The MIT-BIH database contains labels for each beat in a patient trace. The CPSC 2018 database contains labels across complete traces. There were 100 traces extracted from the MIT-BIH database, 50 of which contained PVC beats while 50 contained no PVC beats. A total of 1521 traces were extracted from the CPSC 2018 database, with 604 being labelled as having PVC beats and 917 being labelled as having no PVC beats.

The results are tabulated in Table II and charted in Figure 10 with existing work included for comparison.

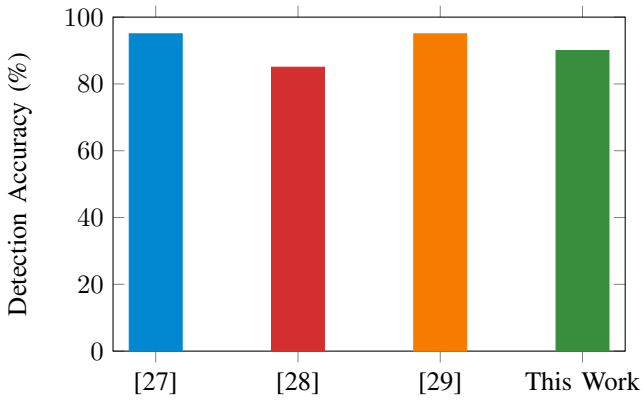


Fig. 10: Charted Premature Ventricular Contraction classification accuracy of this work compared to similar existing work

TABLE II: Premature Ventricular Contraction comparison

Database	Source	Accuracy	Sens.	Spec.
MIT-BIH	This work	90 %	88 %	92 %
MIT-BIH	[27]	95 %	94 %	-
MIT-BIH N	[28]	85 %	98 %	-
MIT-BIH PVC	[28]	85 %	77 %	-
MIT-BIH	[29]	95 %	-	-
CPSC 2018	This work	82 %	68 %	90 %
CPSC 2018 N	[28]	98 %	99.6 %	-
CPSC 2018 PVC	[28]	92 %	82 %	-

Our approach for PVC detection was 90 % accurate, 88 % sensitive, and 92 % specific on the MIT-BIH traces. The results for CPSC 2018 were lower with 82 % accuracy, 68 % sensitivity, and 90 % specificity. The lower sensitivity and higher specificity suggests this method for PVC detection is more suited to diagnosis than screening.

When comparing to other work using the MIT-BIH dataset, our accuracy and sensitivity were placed in the middle, being better than [28] but worse than [27] and [29]. Unfortunately, none of these pieces of work reported on their specificity, making comparisons difficult.

B. Ventricular Tachycardia

VT was assessed using a subset of the MIT-BIH [19] and AAEL databases [24]. Both databases were labelled with timestamped periods of sinus rhythm and a range of arrhythmia. The subsets extracted were those traces labelled with periods of VT and sinus rhythm.

There were 100 traces extracted from the MIT-BIH database, with 36 being labelled as having VT and 64 as sinus rhythm. The AAEL subset had 72 relevant traces, only 9 of which were labelled as having VT, leaving 63 as sinus rhythm.

The results are tabulated in Table III and charted in Figure 11 with existing work included for comparison.

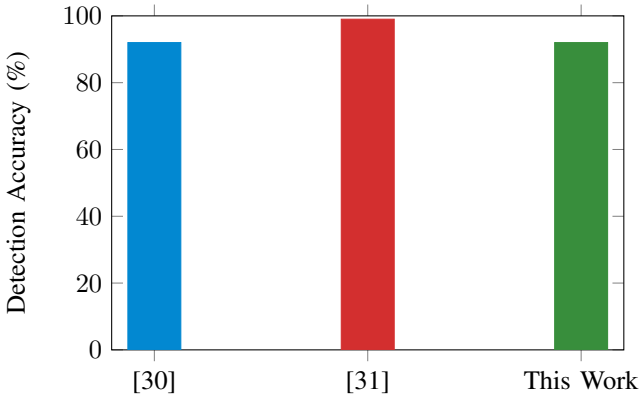


Fig. 11: Charted Ventricular Tachycardia classification accuracy of this work compared to similar existing work

TABLE III: Ventricular Tachycardia comparison

Database	Source	Accuracy	Sens.	Spec.
MIT-BIH Subset	This work	92 %	83 %	97 %
AAEL Subset	This work	79 %	89 %	78 %
20 Recordings	[30]	92 %	-	-
200 Recordings	[31]	99 %	-	-

The detection in the MIT-BIH subset was 92 % accurate, 83 % sensitive, and 97 % specific. Performance on the AAEL subset was overall lower with 79 % accuracy and 78 % specificity, however sensitivity was higher at 89 %. One potential reason for the AAEL subset results being lower is that the dataset was both comparatively small and heavily weighted towards sinus rhythm traces, with only 12.5 % being positive for VT.

Comparison with existing work for this arrhythmia is challenging as the clinical recordings used in [30] and [31] were not able to be acquired, though our accuracy on the MIT-BIH subset is comparable.

C. Atrial Fibrilation

The AF detection was assessed using a subset of the MIT-BIH AF database [22] and the CPSC 2018 training database [23]. Traces selected were those labelled as normal or showing AF.

The MIT-BIH AF traces were based on cardiologist identified periods of sinus rhythm and AF. Each trace contains timestamped labels to identify changes in rhythm, allowing a number of trace subsets to be extracted for each patient as they switched between both states. A total of 100 traces were extracted, 50 of which were labelled as AF and 50 as sinus rhythm.

The CPSC 2018 traces were labelled per trace, and traces which displayed multiple arrhythmia were given up to three labels but were not timestamped. For this work, only complete traces with a single arrhythmia label were used. A total of 1894 traces were extracted, with 976 being labelled as having AF and 918 as being sinus rhythm.

The results are tabulated in Table IV and charted in Figure 12 with existing work included for comparison.

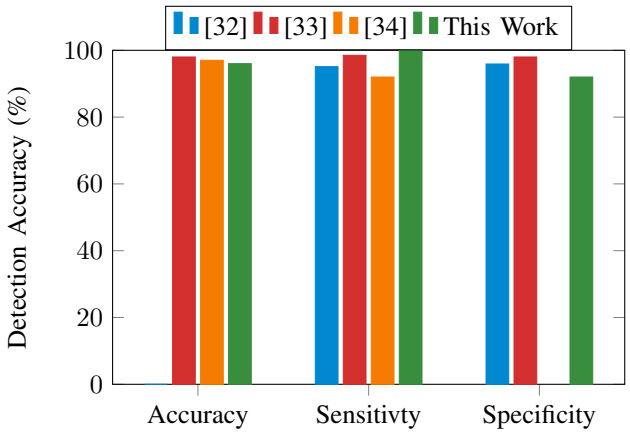


Fig. 12: Charted Atrial Fibrilation classification accuracy, sensitivity, and specificity of this work compared to similar existing work

TABLE IV: Atrial Fibrilation comparison

Database	Source	Accuracy	Sens.	Spec.
CPSC 2018	This work	92.5 %	92 %	93 %
MIT-BIH AF	This work	96 %	100 %	92 %
MIT-BIH AF	[32]	-	95.1 %	95.9 %
MIT-BIH AF	[33]	98 %	98.5 %	98 %
MIT-BIH AF	[34]	97 %	92 %	-

Our approach for AF detection correctly classified 96 % of the MIT-BIH AF traces with 100 % sensitivity and 92 % specificity. Having such a high sensitivity suggests that this approach could be well suited to a screening test. These results also translated well to the CPSC 2018 dataset, with above 90 % results for each of accuracy, sensitivity, and specificity suggesting the approach for AF detection is not specific to a single dataset and is therefore more likely to perform well across a population.

Compared to existing work using the MIT-BIH AF database our accuracy seems comparable. Interestingly, however, our

work seems to have a higher sensitivity (at the cost of specificity), while other work either seems to target a high specificity ([34]) or a balanced sensitivity and specificity ([32] and [33]).

VI. Discussion

A. Explainability

Existing work in arrhythmia classification uses informal specifications. Often these approaches are configured by a training process which results in black box systems. This means the system's behaviour is not explainable and is limited to the information extracted from the training set, making it less useful in a clinical setting. The white box formal expression of our arrhythmia models mitigates this and allows clinical interpretation.

Our approach can be used to model any arrhythmia which can be expressed as a set of policies modelled as TAs. This can capture timing and even basic morphology of ECGs. Much of the existing work focuses on a single arrhythmia, where we have demonstrated our approach performs similarly to a number of these existing approaches across a range of datasets and arrhythmia while still maintaining explainability.

B. Existing Work

Of the existing work in arrhythmia classification we have compared our work to similar existing work. In Table V we provide a comparison of the existing work in relation to this work, categorising the arrhythmia captured, whether the model is formal (white box), whether it requires training, and if the approach is able to be executed in real time.

TABLE V: Comparison of Existing Work

Source	Arrhythmia	Formal	Training	Real Time
[27]	PVC	No	Patient-specific	No
[28]	PVC	No	Yes	Yes
[29]	PVC	No	No	Yes
[30]	VT	No	No	No
[31]	VT	No	Yes	No
[32]	AF	No	Patient-specific	Yes
[33]	AF	No	Yes	Yes
[34]	AF	No	No	Yes
This Work	PVC VT AF	Yes	No	Yes

This table shows that our framework is unique in its approach in that it is the only formally backed system that is white box in nature. Furthermore, of each of these works only two ([28] and [33]) are both able to be run in real time and require no training. Next, we will look at each of these pieces of work in more detail, grouped by the arrhythmia they aim to classify.

1) Premature Ventricular Contraction: Allami et al. [27] use the Pan Tompkins algorithm [35] over three minutes and forecast the QRS count. They use the mean of observed R intervals to set a threshold. This produces a real time final model but is hindered by three minutes of required patient-specific training.

Cai et al. [28] also use the Pan Tompkins algorithm for QRS detection. They calculate an approximate QRS width and height using thresholds based on training data to classify PVCs. As training data is often limited by bias, some QRS complexes not seen in the training data could be misclassified. Additionally, this approach does not consider the P wave.

Chang et al. [29] use wavelets to detect QRS complexes, calculate the mean time between R peaks and uses a threshold to detect early R peaks. This approach only considers the R wave timing, missing the clinical indicators of morphology and missing P waves.

2) Ventricular Tachycardia: Golzar et al. [30] use the Pan Tompkins algorithm for QRS detection to collect R intervals over a minute and compare these to a set threshold. The system takes 70 seconds to process this.

Chowdhury [31] uses a fuzzy approach to calculate entropy which is input to a neural network to detect QRS complexes. Over 60 seconds the QRS rate and width is averaged and compared to a threshold for classifying a trace as VT.

Any period of VT lasting over 30 seconds is considered an emergency [15]. In both pieces of work the processing delay is too long to detect VT that requires emergency intervention.

3) Atrial Fibrilation: Jeon et al. [32] use the Daubechies 8 wavelet to detect the QRS complex and extract features to capture morphology. They use a support vector machine over a 10 s sliding window to detect AF. This informal approach is limited by the requirement of patient-specific training.

Marsili et al. [33] use the entropy of a Finite State Machine (FSM) trace to classify ECG traces as AF in real time on a wearable device. This work considered only the R-R interval for AF detection, excluding the clinical feature of missing P waves.

Babaeizadeh et al. [34] use Markov chains to capture the probability of R wave intervals being AF in real time. This work excluded signal processing which could increase computation time beyond real time deadlines and in practical application would require an external signal processing module.

C. Signal Processing

Careful analysis of ECG traces incorrectly classified by our system shows incorrect signal processing, specifically erroneous wave labelling, to be the most common problem. Our focus in the presented work was on the development of specifications to capture arrhythmia and signal processing is not a key contribution of this work, however, we have invested time in improving the signal processing. This still remains a limitation of the system and additional improvements (specifically developing criteria for detection of erroneous labels) would further improve results.

VII. Conclusions

We have developed a set of Timed Automaton (TA)-based arrhythmia policies based on clinical interpretation of ECGs for classifying Premature Ventricular Contractions (PVCs), Ventricular Tachycardia (VT) and Atrial Fibrilation (AF). Such an approach is based on formal models, allowing the overall system to be white box in nature and improve its

clinical relevance. We synthesised runtime monitors from the arrhythmia policies that, combined with an existing signal processing module, allowed us to evaluate our approach.

Through existing clinical ECG databases we then evaluated our approach in terms of accuracy, sensitivity, and specificity across each of the arrhythmia being looked at. Our approach generally shows comparable accuracy to existing methods when compared to similar data sets. Additionally, our work shows high specificity when detecting VT, making it well suited as a diagnostic tool, and high sensitivity when detecting AF, making it a good fit as a screening test.

As noted in the discussions, one limiting factor of this work is the signal processing, which we hope to improve in the future. Additionally, converting the signal processing in an embedded friendly manner will enable the creation of a portable device, such as a smart watch or stand alone sensor, that can classify arrhythmia on demand and in real time. This approach can also be extended to classify additional arrhythmia that can be represented as formal TA-based policies. This is promising as we work toward the development of a complete embedded system to perform real time classification of ECGs.

REFERENCES

[1] "Cardiovascular diseases cvds." [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] B. Dicker, V. Oliver, G. Howie, J. Bu, G. Stewart, A. Swain, and M. Morris, "Out-of-hospital cardiac arrest registry rēhita mate manawa mō waho i te hōhipera," 2020.

[3] G. Elkilany, R. B. Singh, E. Adeghate, J. Singh, K. Bidasee, O. Shehab, and K. Hristova, "Sudden cardiac death," pp. 51–62, 8 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK507854/

[4] S. Barro, M. Fernandez-Delgado, J. Vila-Sobrino, C. Regueiro, and E. Sanchez, "Classifying multichannel ecg patterns with an adaptive neural network," *IEEE Engineering in Medicine and Biology Magazine*, vol. 17, no. 1, pp. 45–55, 1998.

[5] Z. Dokur, T. Olmez, M. Korurek, and E. Yazgan, "Detection of ecg waveforms by using artificial neural networks," in *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 3. IEEE, 1996, pp. 929–930.

[6] N. Maglaveras, T. Stamkopoulos, C. Pappas, M. Strintzis, T. Stamkopoulost, C. Pappasf, and M. Strintzis, "Ecg processing techniques based on neural networks and bidirectional associative memories," *Journal of Medical Engineering & Technology*, vol. 22, pp. 106–111, 1998.

[7] G. Vijaya, V. Kumar, and H. Verma, "Ann-based qrs-complex analysis of ecg," *Journal of medical engineering & technology*, vol. 22, no. 4, pp. 160–167, 1998.

[8] J. Wing, "A specifier's introduction to formal methods," *Computer*, vol. 23, no. 9, pp. 8–22, 1990.

[9] A. Panda, S. Pinisetty, P. Roop, A. B. Kambhampati, and M. S. Manikandan, "Runtime verification of implantable medical devices using multiple physiological signals," *ACM Symposium on Applied Computing*, Mar. 2021.

[10] H. Pearce, M. Kuo, P. Roop, and S. Pinisetty, "Securing implantable medical devices with runtime enforcement hardware," in *Proceedings of the 17th ACM-IEEE International Conference on formal methods and models for system design*, ser. MEMOCODE '19. ACM, 2019, pp. 1–9.

[11] J. Wang, S. Liu, Y. Qi, and D. Hou, "Developing an insulin pump system using the sofl method," in *14th Asia-Pacific Software Engineering Conference (APSEC'07)*, 2007, pp. 334–341.

[12] S. Pinisetty, T. Jéron, S. Tripakis, Y. Falcone, H. Marchand, and V. Preoteasa, "Predictive runtime verification of timed properties," *Journal of Systems and Software*, vol. 132, pp. 353–365, 2017.

[13] Y. Falcone, J. C. Fernandez, and L. Mounier, "Runtime verification of safety-progress properties," vol. 5779 LNCS, 2009, pp. 40–59.

[14] R. Alur and D. L. Dill, "A theory of timed automata," 1990.

[15] M. S. Thaler, *The Only EKG BOOK You'll Ever Need NINTH EDITION*, 2019.

[16] S. Pinisetty, T. Jéron, S. Tripakis, Y. Falcone, H. Marchand, and V. Preoteasa, "Predictive runtime verification of timed properties," *Journal of Systems and Software*, vol. 132, 06 2017.

[17] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304397594900108

[18] D. A. Warrell, E. J. Benz Jr, T. M. Cox, and J. D. Firth, *Oxford textbook of medicine*. Oxford University Press, USA, 2003.

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

[20] 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, 2000.

[21] S. Pinisetty, P. S. Roop, V. Sawant, and G. Schneider, "Security of pacemakers using runtime verification," *2018 16th ACM/IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE 2018*, pp. 1–11, 2018.

[22] G. Moody, "A new method for detecting atrial fibrillation using rr intervals," *Computers in Cardiology*, pp. 227–230, 1983.

[23] F. Liu, C. Liu, L. Zhao, X. Zhang, X. Wu, X. Xu, Y. Liu, C. Ma, S. Wei, Z. He, J. Li, E. Ng, and Y. Kwee, "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection," *Journal of Medical Imaging and Health Informatics*, vol. 8, pp. 1368–1373, 2018. [Online]. Available: http://www.icbeb.org/Challenge.html

[24] J. Jenkins and R. Jenkins, "Arrhythmia database for algorithm testing: Surface leads plus intracardiac leads for validation," *Journal of electrocardiology*, vol. 36 Suppl, pp. 157–61, 02 2003.

[25] N. Pilia, C. Nagel, G. Lenis, S. Becker, O. Dössel, and A. Loewe, "Ecgdeli - an open source ecg delineation toolbox for matlab," *SoftwareX*, 2021.

[26] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi, "Uppaal—a tool suite for automatic verification of real-time systems," in *Proceedings of the DIMACS/SYCON Workshop on Hybrid Systems III : Verification and Control: Verification and Control*. Berlin, Heidelberg: Springer-Verlag, 1996, p. 232–243.

[27] R. Allami, A. Stranieri, V. Balasubramanian, and H. F. Jelinek, "A count data model for heart rate variability forecasting and premature ventricular contraction detection," *Signal, Image and Video Processing*, 2017.

[28] Z. Cai, J. Li, A. E. Johnson, X. Zhang, Q. Shen, J. Zhang, and C. Liu, "Rule-based rough-refined two-step-procedure for real-time premature beat detection in single-lead ecg," *Physiological Measurement*, 2020.

[29] R. C.-H. Chang, C.-H. Lin, M.-F. Wei, K.-H. Lin, and S.-R. Chen, "High-precision real-time premature ventricular contraction (pvc) detection system based on wavelet transform," *Journal of Signal Processing Systems*, 2014.

[30] M. Golzar, F. Fotouhi-Ghazvini, H. Rabbani, and F. SadatZakeri, "Mobile cardiac health-care monitoring and notification with real timetachycardia and bradycardia arrhythmia detection," *Journal of Medical Signals & Sensors*, 2017.

[31] S. Roy Chowdhury, "High-resolution detection of sustained ventricular and supraventricular tachycardia through fpga-based fuzzy processing of ecg signal," *Medical and Biological Engineering and Computing*, 2015.

[32] T. Jeon, B. Kim, M. Jeon, and B. G. Lee, "Implementation of a portable device for real-time ecg signal analysis," *BioMedical Engineering Online*, vol. 13, 2014.

[33] I. A. Marsili, L. Biasiolli, M. Masè, A. Adami, A. O. Andrighetti, F. Ravelli, and G. Nollo, "Implementation and validation of real-time algorithms for atrial fibrillation detection on a wearable ecg device," *Computers in Biology and Medicine*, vol. 116, p. 103540, 1 2020.

[34] S. Babaeizadeh, R. E. Gregg, E. D. Helfenbein, J. M. Lindauer, and S. H. Zhou, "Improvements in atrial fibrillation detection for real-time monitoring," *Journal of Electrocardiology*, vol. 42, pp. 522–526, 11 2009.

[35] J. Pan and W. J. Tompkins, "Real-time qrs detection algorithm." *IEEE Transactions on Biomedical Engineering*, vol. BME-32, pp. 230–236, 1985.