Extension and Detailed Overview of the HRVFrame Framework for Heart Rate Variability Analysis

Alan Jovic [#1], Nikola Bogunovic [#2], Marko Cupic [#3]

[#] *Department of Electronics, Microelectronics, Computer and Intelligent Systems, Faculty of Electrical Engineering and Computing, University of Zagreb*
Unska 3, HR-10000 Zagreb, Croatia
[1] alan.jovic@fer.hr, [2] nikola.bogunovic@fer.hr, [3] marko.cupic@fer.hr

Abstract—**The analysis of heart rate variability (HRV) is an important diagnostic method for detection and assessment of cardiac abnormalities. The availability of complete computer frameworks that can aid researchers in the field of HRV analysis is limited due to the large number of different feature extraction methods. A recently developed framework for feature extraction from cardiac rhythm called HRVFrame is promising, because it allows a user to access more than 40 implemented linear and nonlinear methods. The aim of this paper is to provide a more detailed overview of this framework and all of its capabilities and recent extensions. Additionally, the aim is to encourage the use of HRVFrame as a free and open-source tool for developing medical applications based on Java programming language. A comparison of the framework with existing solutions for cardiac rhythm analysis is provided.**

Keywords: feature extraction, heart rate variability, framework, Java programming, open-source software, medical applications

I. Introduction

Feature extraction from biomedical time-series (BTS) is a large and ever expanding field of scientific research. BTS tend to behave in often unpredictable or only somewhat predictable ways because of the complexity of the underlying physiology [1]. Cardiac rhythm, as a specific type of BTS, shows interesting short- and long-scale complex fluctuations [2]. Aside from describing the very nature of these fluctuations, they can be analyzed in order to: model cardiac disorders [3], classify cardiac rhythms [4], predict the onset of cardiac disorders [5], and establish disorder prognosis [6]. The analysis of the fluctuations of the cardiac inter-beat intervals is known as heart rate variability (HRV) analysis. The significance of HRV analysis lies not only in the description and detection of cardiac disorders, but also in the assessment of other states and disorders of the human body. Herein, cardiac rhythm can be only one of several BTS that contribute to disorder description [7].

Scientific research in biomedicine relies on the ability to easily and efficiently compare the results of one's own work with that of others. The comparison can be severely hindered due to several reasons, such as the use of non-referential datasets, the analysis of different types of disorders and the lack of access to the implementations of other researchers' work. With the latter issue in mind, in a recent paper, we have described a novel Java-based computer framework for HRV analysis called HRVFrame [8]. The main goal of this framework is to allow the researchers in the domain of BTS analysis to access a large number of existing linear and nonlinear methods available in this field.

The goal of this paper is to provide a more detailed insight into the structure of the framework and also to mention several extensions to the framework from earlier versions. We aim to encourage the wide-spread application of the framework in BTS analysis. The previous versions of the framework were already provided to researchers from several countries upon their request. The framework is now available as open-source software under General Public License at: http://www.zemris.fer.hr/~ajovic/hrvframe/hrvframe.html.

The paper is organized as follows. In Section II, we provide a detailed overview of the framework from the functional perspective. Section III deals with the implemented methods and their use. In Section IV, we describe the framework parameters and the application to HRV analysis. Section V compares the framework with some of the existing solutions for HRV analysis. Section VI concludes the paper.

II. HRVFrame Functionality

HRVFrame is an extensive Java-based framework containing many features covered in the HRV analysis literature. The process of feature extraction is currently performed offline and the framework is a stand-alone product. It is not integrated with any particular knowledge-based platform. The previous versions of the framework were successfully used for cardiac disorder classification based on HRV analysis [9] as well as for evaluation of the best feature combinations for cardiac arrhythmia classification based on cardiac rhythm alone [10].

A. Types of usage

HRVFrame framework can be used in two ways. The first way is to use the whole framework, which includes both the core packages that contain feature extraction methods as well as additional packages for graphical user interface (GUI), data input, and feature vectors output. This usage mode enables batch feature extraction from cardiac rhythm records. It allows recording of output feature vectors in a format suitable for further analysis – data mining in a knowledge discovery platform (.arff format). This process is shown in Fig. 1.

The other way is to use only the core packages of the framework in a client's medical application. In this mode, the

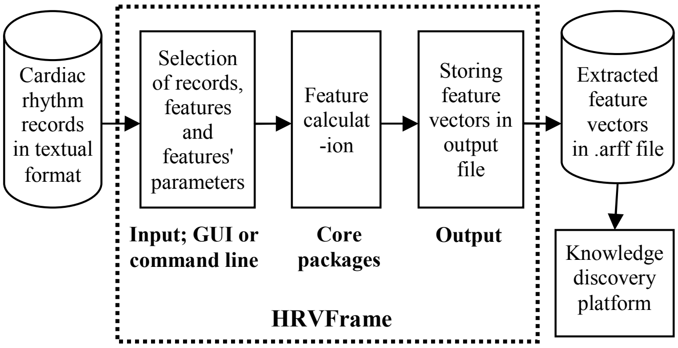

Cardiac rhythm records in textual format
Selection of records, features and features' parameters
Feature calculat -ion
Storing feature vectors in output file
Extracted feature vectors in .arff file
Knowledge discovery platform
Input; GUI or command line
Core packages
Output
HRVFrame

Fig. 1. The use of the whole HRVFrame framework for feature extraction and feature vectors' preparation for data mining, adapted from [8].

client has to implement all of the other Java classes that deal with data input and output in order to solve his problem. The framework's core packages can work with any type of BTS, and only some of the standard linear and a few nonlinear methods are specific for cardiac rhythm analysis, as described in the guidelines for HRV analysis [11] and in Section III. An example of the second way of usage of the framework is shown in Fig. 2. In this example, the core packages of the framework are used to analyze both the cardiac rhythm and electroencephalogram from the same patient.

B. GUI and command line usage

The framework's GUI allows access to all of the implemented methods of the framework with their array of possible parameters. Through the use of GUI, the user can quickly select input cardiac records, specify various types of analyses, select the features that are to be extracted, and set their parameters (if any). The output file in .arff format can also be created or selected for appending feature vectors. The extraction process can be started once the input file(s), output file and features are defined. An example of the GUI window for selection of the nonlinear features is shown in Fig. 3.

Instead of GUI, the framework can also be controlled using command-line interface, with the same capabilities as those of the GUI. The general command-line call to the framework is:

```
>> java -jar HRVFrame.jar <output_file>
<analysis_parameters> <features_and_their_parameters>
-use <list_of_input_files>
```

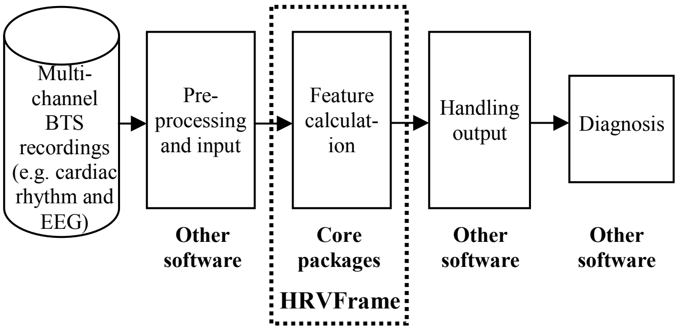
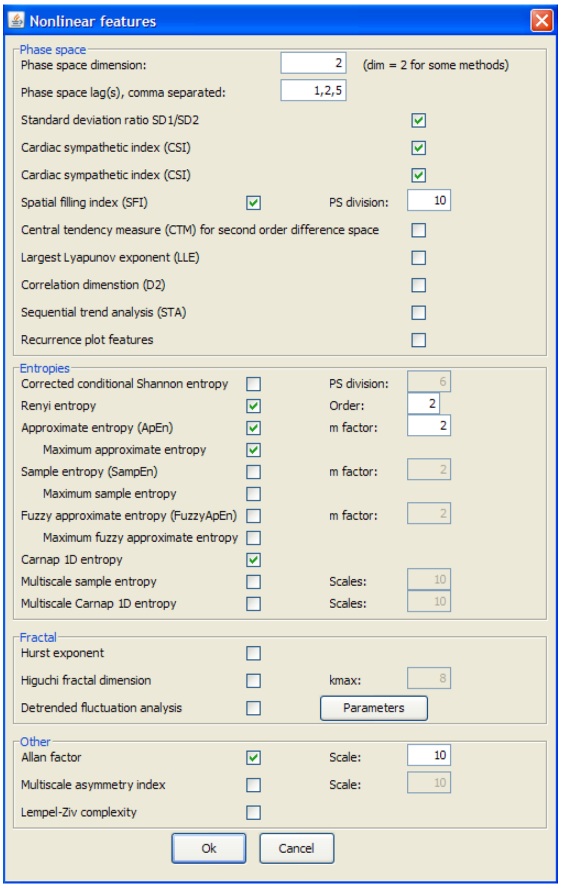


Fig. 2. An example of the use of core packages of the HRVFrame framework for integration with other BTS and other medical software.



Phase space
Phase space dimension: 2 (dim = 2 for some methods)
Phase space lag(s), comma separated: 1,2,5
Standard deviation ratio SD1/SD2 ☑
Cardiac sympathetic index (CSI) ☑
Cardiac sympathetic index (CSI) ☑
Spatial filling index (SFI) ☑ PS division: 10
Central tendency measure (CTM) for second order difference space ☐
Largest Lyapunov exponent (LLE) ☐
Correlation dimenstion (D2) ☐
Sequential trend analysis (STA) ☐
Recurrence plot features ☐

Entropies
Corrected conditional Shannon entropy ☐ PS division: 6
Renyi entropy ☑ Order: 2
Approximate entropy (ApEn) ☑ m factor: 2
Maximum approximate entropy ☑
Sample entropy (SampEn) ☐ m factor: 2
Maximum sample entropy ☐
Fuzzy approximate entropy (FuzzyApEn) ☐ m factor: 2
Maximum fuzzy approximate entropy ☐
Carnap 1D entropy ☑
Multiscale sample entropy ☐ Scales: 10
Multiscale Carnap 1D entropy ☐ Scales: 10

Fractal
Hurst exponent ☐
Higuchi fractal dimension ☐ kmax: 8
Detrended fluctuation analysis ☐ Parameters

Other
Allan factor ☑ Scale: 10
Multiscale asymmetry index ☐ Scale: 10
Lempel-Ziv complexity ☐

Fig. 3. Graphical user interface of the framework for specifying nonlinear features.

In this call, the order of arguments is not important. However, the argument `-use` and the list of input files need to be put at the end of the arguments list.

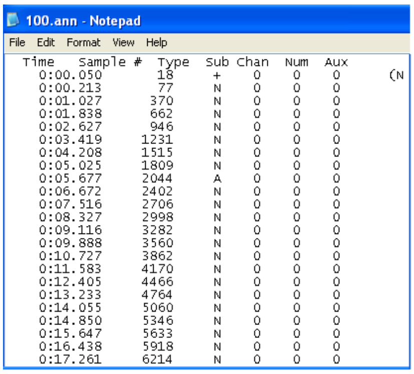
Here is an example for creating an output file named `hrv_research.arff`:

```
> java -jar HRVFrame.jar -carffhrv_research.arff -
relHRV_research_data -rec -st -co -sfi -lags1 -hur
```

This will create the output file `hrv_research.arff` with the name of the relation `HRV_research_data`, with the name of the record as an attribute (`-rec`), with attributes for the ordinal number of the starting annotation (`-st`) and the segment length (`-co`, in the number of annotations) included. It will also contain two attributes for features: the spatial filling index with lag=1 (`-lags1`) and the Hurst exponent (`-hur`). The information about the phase space lag is required for proper labeling of the spatial filling index feature in the output file. Omitting it would result in an error reported by the framework. The information about all of the possible command line parameters for the framework can be obtained by using the parameter `-help`.

C. Data input

The framework contains an input class that loads cardiac rhythm records. The acceptable input is in the form of textual



```
  Time    Sample #  Type   Sub Chan  Num   Aux
    0:00.050       18     +    0    0    0         (N
    0:00.213       77     N    0    0    0
    0:01.027      370     N    0    0    0
    0:01.838      662     N    0    0    0
    0:02.627      946     N    0    0    0
    0:03.419     1231     N    0    0    0
    0:04.208     1515     N    0    0    0
    0:05.025     1809     N    0    0    0
    0:05.677     2044     A    0    0    0
    0:06.672     2402     N    0    0    0
    0:07.516     2706     N    0    0    0
    0:08.327     2998     N    0    0    0
    0:09.116     3282     N    0    0    0
    0:09.888     3560     N    0    0    0
    0:10.727     3862     N    0    0    0
    0:11.583     4170     N    0    0    0
    0:12.405     4466     N    0    0    0
    0:13.233     4764     N    0    0    0
    0:14.055     5060     N    0    0    0
    0:14.850     5346     N    0    0    0
    0:15.647     5633     N    0    0    0
    0:16.438     5918     N    0    0    0
    0:17.261     6214     N    0    0    0
```

Fig. 4. An example of the correct input format for the cardiac rhythm records that are analyzed by HRVFrame.

ASCII files. The input file structure is the same as the one provided by the PhysioNet tool *rdann* [12]. The files should contain the information on the times of R peaks, types of beats, and optional rhythm annotations. An example of the correct format is shown in Fig. 4. Most of the beat and rhythm annotations used in PhysioNet's electrocardiogram and cardiac rhythm databases are supported by the framework. Also acceptable are the input formats where only R peak times are known or where the beat types are known, but not the underlying rhythm.

D. Feature vectors' output

The framework allows users to first create an .arff file that only contains the declaration of features (attributes in .arff file format). After the feature calculation process for an RR segment is complete, the framework creates output feature vectors and stores them in the .arff file. Data are appended to the end of the file as long as the new segments and records are continued to be analyzed. The .arff file can then be used by several data mining and knowledge discovery platforms such as Weka [13] and RapidMiner [14].

III. Implemented Methods

HRVFrame implements more than 40 linear and nonlinear methods for feature extraction from BTS, with some methods capable of extracting several particular features. Also, some methods have parameters that determine the features' calculation procedure. Most of the implemented methods are non-specific for HRV analysis and can be used for calculation of entropy, fractal, chaotic, and other properties of any BTS. The methods that intend to be used in the analysis have to be specified by the researcher, for currently there is no intelligent selection of appropriate features with respect to the analyzed disorders or segment lengths, mainly because this area is still under research. A thorough search of the available literature was performed that resulted in implementation of most of the methods applied in HRV research. The features are calculated in batch for one record segment at a time.

The structure of the core packages of the framework is presented in Table I. Some of the packages (e.g. features.linear.frequencyDomain.operations, statisticMeasure) contain classes with methods needed for other classes in the framework to work properly, so the core packages are meant to be used together.

The names of the packages and classes were not chosen by any guidelines. The Task Force responsible for creating the HRV guidelines mentions only time-domain methods, frequency-domain methods, rhythm pattern analysis and non-linear methods for HRV, but does not suggest any standard classification or labeling of the existing methods [11]. A recent review of the variability methods for BTS analysis in general argues for the classification of variability methods into statistical, geometric, energetic, informational, and invariant [15]. Other classifications of the methods is also possible such

TABLE I

The Structure of the Core Packages of the Framework

Package	Class
features.linear.frequencyDomain	SpectralAnalysis
	SpectralEntropy
features.linear.frequencyDomain.operations	Complex
	FFT
	RandomPhases
	SurrogateSeries
features.linear.timeDomain	AutocorrelationCoefficient
	FanoFactor
	HRVTriangularIndex
	PNNX
	RMSSD
	SDANN
	SDNN
	SDSD
	TINN
features.nonlinear.entropy	ApEn
	CarnapEntropy1D
	CorrectedConditionalShannonEntropy
	FuzzyApEn
	RenyiEntropy
	SampEn
features.nonlinear.fractal	DFA
	FractalDimension
	HiguchiDimension
	HurstExponent
features.nonlinear.other	AllanFactor
	LempelZivComplexity
	MultiscaleAsymmetryIndex
features.nonlinear.phaseSpace	CorrelationDimension
	CTM
	LyapunovExponent
	ParametersTakens
	RecurrencePlot
	SequentialTrendAnalysis
	SpatialFillingIndex
	StandardDeviationsRatio
features.symbolicDynamics	SequentialSpectrum
features.timeFrequency	HaarWaveletStandardDeviation
	HilbertHuangTransform
statisticMeasure	DistanceMeasure
	Statistics

as: time, frequency, entropy, and scale-invariant [16]. The primary intention of the framework is to provide the implementation of the useful methods for BTS analysis, not to propose or dictate any classification of the methods. Detailed overview of the implemented methods is given in Table II. It should be mentioned that the implementation and testing of all

TABLE II

METHODS IMPLEMENTED IN THE HRVFRAME FRAMEWORK

Method or single feature name	Framework class with method's implementation	Category	Invocation type	Extension	Implemented according to
Mean RR interval [*]	Statistics	(1)	static	no	[11]
SDNN [*]	SDNN	(1)	static	no	[11]
RMSSD [*]	RMSSD	(1)	static	no	[11]
SDSD [*]	SDSD	(1)	static	no	[11]
pNN*X* [*]	PNNX	(1)	static	no	[11]
SDANN [*]	SDANN	(1)	static	no	[11]
Fano factor	FanoFactor	(1)	static	no	[17]
HTI [*]	HRVTriangularIndex	(1)	static	no	[11]
TINN [*]	TINN	(1)	static	no	[11]
ULF [*#]	SpectralAnalysis	(2)	instantiation	no	[11], [18], [19]
VLF [*#]	SpectralAnalysis	(2)	instantiation	no	[11], [18], [19]
LF [*#]	SpectralAnalysis	(2)	instantiation	no	[11], [18], [19]
HF[*#]	SpectralAnalysis	(2)	instantiation	no	[11], [18], [19]
LF/HF[*#]	SpectralAnalysis	(2)	instantiation	no	[11], [18], [19]
Total PSD[#]	SpectralAnalysis	(2)	instantiation	no	[11], [18], [19]
Allan factor	AllanFactor	(8)	static	no	[17]
Autocorrelation coefficient	AutocorrelationCoefficient	(1)	static	yes	[17]
Spatial filling index	SpatialFillingIndex	(4)	instantiation	no	[20]
SD1/SD2 [*]	StandardDeviationsRatio	(4)	instantiation	no	[21]
CSI and CVI [*]	StandardDeviationsRatio	(4)	instantiation	no	[21]
Correlation dimension	CorrelationDimension	(4)	static	no	[22]
Largest Lyapunov Exponent	LyapunovExponent	(4)	static	no	[23]
Central tendency measure	CTM	(4)	static	no	[24]
Sequential trend analysis	SequentialTrendAnalysis	(4)	static	no	[25]
Spectral entropy	SpectralEntropy	(6)	static	no	[26]
Corrected conditional entropy	CorrectedConditionalEntropyShannon	(6)	static	yes	[27]
Approximate entropy (also maximum ApEn)	ApEn	(6)	static	no	[28]
Sample entropy (also maximum SampEn)	SampEn	(6)	static	no	[29]
Multiscale sample entropy	SampEn	(6)	static	no	[30]
Rényi entropy	RenyiEntropy	(6)	static	no	[31]
Fuzzy ApEn	FuzzyApEn	(6)	static	yes	[32]
Carnap 1D entropy	CarnapEntropy1D	(6)	static	yes	[33]
Multiscale Carnap 1D entropy	CarnapEntropy1D	(6)	static	yes	[30],[33]
Multiscale asymmetry index	MultiscaleAsymmetryIndex	(8)	static	no	[34]
Lempel-Ziv complexity	LempelZivComplexity	(8)	static	yes	[35]
Sequential spectrum	SequentialSpectrum	(7)	instantiation	yes	[36]
Higuchi's fractal dimension	HiguchiDimension	(5)	instantiation	no	[37]
Hurst exponent	HurstExponent	(5)	static	no	[17]
Detrended fluctuation analysis	DFA	(5)	static	no	[38]
Haar wavelet's standard deviation	HaarWaveletStandardDeviation	(3)	static	no	[17]
Hilbert-Huang transform	HilbertHuangTransform	(3)	instantiation	yes	[39]
Recurrence plot	RecurrencePlot	(4)	instantiation	yes	[40]

[*] A method or single feature intended for HRV analysis only.

[#] Frequency-domain features can be extracted using non-parametric (FFT+window) or parametric (Burg autoreggresive model) methods [11].

Category: (1) Linear, time-domain; (2) Linear, frequency-domain; (3) Time-frequency; (4) Nonlinear, phase space; (5) Nonlinear, fractal; (6) Nonlinear, entropy; (7) Symbolic dynamics; (8) Nonlinear, other.

the methods in Java took several years of research into this field. Table II provides details on the exact literature from which the methods were implemented, even if the methods were first described in an earlier work. Some of the methods require that the corresponding Java class is instantiated before the extraction can begin. Usually, such classes have several options for extracting the features. Other, simpler methods use static invocation to extract the features. The novel extensions of the framework from previously published work [8] are also listed in Table II. These include: autocorrelation coefficient [17], corrected conditional (Shannon) entropy [27], fuzzy approximate entropy [32], single scale and multiscale Carnap 1D entropy [30],[33], Lempel-Ziv complexity[35], sequential spectrum [36], Hilbert-Huang transform for empirical mode decomposition [39], and recurrence plot [40].

IV. Framework Parameters and HRV Analysis

The use of the whole framework is specifically designed for HRV analysis. The framework supports a number of parameters that specify the exact type of analysis. The parameters can be chosen from GUI or entered through the command line. The list of possible analysis parameters is shown in Table III.

The framework can be used for two types of HRV analysis. The first one is the analysis of cardiac disorders, in which one specifies the type of disorder represented by the cardiac rhythm records given to the framework (-sig<DisType>, Table III). The framework then assumes that the disorder is present in the whole record, which is usually true. The disorder type is considered as the target attribute, having any user-selected label. Any general or specific type of disorder can be studied based on the heart rhythm (e.g. congestive heart failure, cardiac arrhythmia).

The second type of analysis one can pursue is the analysis of the exact rhythm present in the records, analyzed segment by segment (-lfb, Table III). Each segment, depending on its length, can contain a number of different rhythm annotations. Sometimes a segment that is annotated as normal rhythm contains anomalous beats (e.g. premature ventricular contractions, PVC). The rhythm in this segment is not considered normal and is labeled as abnormal depending on the nature of the anomalous beats.

The framework always designates one type of rhythm per segment. When several different annotated rhythms or anomalous beats are encountered, the framework selects the designation based on the rhythm priority. The priorities can be defined in a separate file by the user. Default priority for currently supported abnormalities is shown in Table IV in the descending order. For description of the rhythms and beats the reader is referred to [41]. The priorities of the rhythm level always take precedence over the priorities of the beat level. All abnormalities have higher priority than the normal rhythm. At the moment, sinus tachycardia and sinus arrhythmia are considered to be normal rhythms. Ventricular fibrillation (VF) is currently not considered due to great difficulty in finding significant ventricular responses during VF episodes in surface electrocardiogram.

TABLE III

Input Parameters of the Framework for the HRV Analysis

Analysis parameter	Explanation
-carff<Path>	Create output file named <Path> (path to the file).
-aarff<Path>	Append feature vectors to the file named <Path> (path to the file).
-ointervals	Outputs RR interval durations (tachogram) into a separate text file.
-phspace	Outputs RR interval durations as phase space points of specified dimension (default=2)
-rel<RelName>	Defines relation name in output file. Default relation name is “HRVdata”.
-efr<Path>	Uses file name <Path> that determines rhythm priorities when labeling segments.
-odifferences	Outputs RR interval differences durations to a separate text file.
-lfb	Framework determines rhythm type based on rhythm annotations and cardiac beats.
-qaq	Considers QRS artifacts as valid R peaks.
-dnb	Disregards non-beat annotations when constructing RR interval series.
-test	Considers only R peak times and does not label the rhythm, used for testing.
-er	Segments are analyzed until the end of file.
-rec	Includes record file name as an attribute in the output .arff file (as string).
-start*X*	Extraction starts with *X*-th annotation.
-st	Uses the starting ordinal number of the annotation as an attribute in the output file.
-sttime*X*	Analysis starts at moment *X* seconds after the beginning of the record.
-tis	Uses the starting time as an attribute in the output .arff file.
-count*X*	One analyzed segment includes *X* annotations
-co	Uses the number of annotations in the segment as an attribute in the output .arff file.
-segm*X*	One analyzed segment lasts *X* seconds
-seg	Uses segment length in seconds as an attribute in the output .arff file.
-sig<DisType>	Explicit specification of the disorder present in all of the extracted segments.
-use <FileList>	Uses heart rate records as a list of files (files' paths separated by comma)

TABLE IV

Default Priorities for the Supported Cardiac Rhythm and Beat Abnormalities

Rhythm level	ventricular flutter -> ventricular tachycardia -> ventricular bigeminy -> ventricular trigeminy -> Mobitz II second degree heart block -> idioventricular rhythm -> atrial flutter -> atrial fibrillation -> supraventricular tachyarrhythmia -> AV nodal rhythm -> atrial bigeminy -> atrial trigeminy -> paced rhythm -> sinus pause -> sinus bradycardia
Beat level	Compensated couplet -> decompensated couplet -> PVC -> fusion -> ventricular ectopy -> PVC and PAC -> PAC -> LBBB -> RBBB -> normal

V. COMPARISON TO SIMILAR WORK

There is a growing body of computer software that can extract features from cardiac rhythm. Some of the software is more user-friendly and practice-oriented, mostly towards medical personnel, while other aims more towards biomedical research. In Table V, we present a high-level comparison of the existing software for HRV analysis, with intention to show the advantages and shortcomings of each software product.

The difference between a toolkit and a framework is marginal. Basically, a framework is intended to allow construction of other applications while a toolkit is only intended to bring additional functionality to an already existing platform (e.g. Matlab or R). We estimate the community support level based on two factors: the estimated size of the community that uses the software (the larger the community, the higher the support) and the total number of implemented methods (the larger the number, the more likely it is that the software will be used and developed further).

Compared to HRVFrame, all of the other software contains less implemented methods for HRV and BTS analysis in general, especially the nonlinear methods. The advantage of Matlab-based software (e.g. ECGLab [42], KARDIA [43]) lies in their integration with other functions of the platform, including visualization and advanced signal processing toolboxes, as well as the size of its community. However, Matlab is not a free piece of software which limits its applicability. Stand-alone applications like KUBIOS [45] are useful for medical personnel, but they are limited for scientific explorations. Extensive open-source frameworks like BioSig [44], which is aimed more towards brain-computer interface, have advantages in their scale and supported standards, but are not specialized for HRV analysis.

The implementation of HRVFrame in Java was chosen because of platform independence, easy GUI construction, and general public license of the developed Java code. The code is reliable and fast. The implementations in other languages such as C, Matlab, and R were also considered, but we concluded that the best implementation is the one in Java due to these reasons.

VI. CONCLUSION

In this work, we have shown that HRVFrame is a large and mature framework for HRV analysis. Its implementation in Java as an open-source software allows it to be used and modified on any platform. HRVFrame can be extended further to include the packages for the analysis of other types of BTS. This will the topic of future work.

The main use of the framework is in scientific explorations and comparison of results with similar work. The framework allows the researcher to experiment with his own feature implementation and compare the results with other features on the same datasets in classification or prediction of particular cardiac disorders. Its core packages can also be used in any client's medical application that focuses on feature extraction and data mining from biomedical time-series.

REFERENCES

[1] L. Glass, "Synchronization and rhythmic processes in physiology," *Nature*, vol. 410, no. 6825, pp. 277–284, Mar. 2001.

[2] A. L. Goldberger, "Non-linear dynamics for clinicians: chaos theory, fractal, and complexity at the bedside," *Lancet*, vol. 347, no. 9011, pp. 1312–1314, Jan. 1996.

[3] J. W. Lin, J. J. Hwang, L. Y. Lin, and J. L. Lin, "Measuring Heart Rate Variability with Wavelet Thresholds and Energy Components in Healthy Subjects and Patients with Congestive Heart Failure," *Cardiology*, vol. 106, no. 4, pp. 207–214, Aug. 2006.

[4] B. M. Asl, S. K. Setarehdan, and M. Mohebbi, "Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal," *Artif. Intell. Med.*, vol. 44, no. 1, pp. 51–64, Sep. 2008.

[5] G. Manis, S. Nikolopoulos, A. Alexandridi, and C. Davos, "Assessment of the classification capability of prediction and approximation methods for HRV analysis," *Comp. Biol. Med.*, vol. 37, no. 5, pp. 642–654, May 2007.

[6] H. V. Huikuri, T. H. Makikallio, C. K. Peng, A. L. Goldberger, U. Hintze, and M. Moller, "Fractal correlation properties of R–R interval dynamics and mortality in patients with depressed left ventricular function after an acute myocardial infarction," *Circulation*, vol. 101, no. 1, pp. 47–53, Jan. 2000.

TABLE V

COMPARISON OF HRVFRAME WITH THE EXISTING SOFTWARE FOR HRV ANALYSIS

Software name	Type	Language	Main orient-ation	Availability	Community support level	Implemented feature groups for HRV	Estimated total no. of methods
ECGLab [42]	toolkit	Matlab	research	Matlab license	high	time, frequency, time-frequency, nonlinear	15
KARDIA [43]	toolkit/ application	Matlab	research/ practice	Matlab license	medium	time, frequency, nonlinear	15
BioSig [44]	framework	C/C++, Matlab	research	open-source	medium	time, frequency	10
KUBIOS-HRV [45]	application	Matlab, C	practice	free software, not open-source	high	time, frequency, nonlinear	19
R-HRV [46]	toolkit	C, R	research	open-source, GPL	medium	time, frequency, nonlinear	15
HRV Toolkit [47]	toolkit	C	research	open-source	medium	time, frequency	13
Software for HRV [48]	toolkit	Matlab	research/ practice	free software, not open-source	low	time, frequency, nonlinear	9
HRVFrame	framework	Java	research	open source, GPL	medium	time, frequency, time-frequency, symbolic dynamics, nonlinear	42

[7] K. Ćosić, S. Popović, D. Kukolja, M. Horvat, and B. Dropuljić, Physiology-driven adaptive virtual reality stimulation for prevention and treatment of stress related disorders, *CyberPsychology, Behavior, and Social Networking*, vol. 13, no. 1, pp. 73–78, Feb. 2010.

[8] A. Jovic and N. Bogunovic, "HRVFrame: Java-Based Framework for Feature Extraction from Cardiac Rhythm," *Lecture Notes in Artificial Intelligence*, vol. 6747, pp. 96–100, Jul. 2011.

[9] A. Jovic and N. Bogunovic, "Electrocardiogram analysis using a combination of statistical, geometric, and nonlinear heart rate variability features," *Artif. Intell. Med.*, vol. 51, no. 3, pp. 175–186, Mar. 2011.

[10] A. Jovic and N. Bogunovic, "Evaluating and Comparing Performance of Feature Combinations of Heart Rate Variability Measures for Cardiac Rhythm Classification," *Biomed. Sig. Proc. Control*, vol. 7 no. 3, pp. 245–255, May 2012.

[11] Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology, "Heart rate variability guidelines: Standards of measurement, physiological interpretation, and clinical use," *Eur. Heart J.*, vol. 17, no. 3, pp. 354–381, Mar. 1996.

[12] (2012) PhysioNet tool *rdann*. [Online]. Available: http://www.physionet.org/tutorials/physiobank-text.shtml#ann

[13] I. H. Witten and E. Frank, *Data mining: Practical machine learning tools and techniques*, 3rd ed., San Francisco, USA: Morgan Kaufmann, 2011.

[14] (2012) RapidMiner. [Online]. Available: http://rapid-i.com/content/view/181/190/

[15] A. Bravi, A. Longtin, and A. J. E. Seely, "Review and classification of variability analysis techniques with clinical applications," *BioMedical Engineering OnLine*, vol. 10, p. 90, Oct. 2011.

[16] A. J. E. Seely and P. T. Macklem, "Complex systems and the technology of variability analysis," *Critical Care*, vol. 8, no. 6, pp. 367–384, Sep. 2004.

[17] M. C. Teich, S. B. Lowen, B. M. Jost, K. Vibe-Rheymer, and C. Heneghan, "Heart-Rate Variability: Measures and Models," in M. Akay, Ed., *Dynamic Analysis and Modeling*, ser. Nonlinear Biomedical Signal Processing, New York, USA: IEEE Press, 2001, vol. II, ch. 6, pp. 159–213.

[18] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed., Cambridge, UK: Cambridge University Press, 1992.

[19] S. de Waele and P. M. T. Broersen, "The Burg Algorithm for Segments," *IEEE Trans. Signal Process.*, vol. 48, no. 10, pp. 2876–2880, Oct. 2000.

[20] O. Faust, R. U. Acharya, S. M. Krishnan, and L. C. Min, "Analysis of cardiac signals using spatial filling index and time-frequency domain," *BioMedical Engineering OnLine*, vol. 3, p. 30, Sep. 2004.

[21] C.-W. Lin, J.-S. Wang, and P.-C. Chung, "Mining Physiological Conditions from Heart Rate Variability Analysis," *IEEE Computat. Intell. Mag.*, vol. 5, no. 1, pp. 50–58, Feb. 2010.

[22] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D: Nonlinear Phenomena*, vol. 9, no. 1–2, pp. 189–208, Oct. 1983.

[23] M. Rosenstien, J. J. Colins, and C. J. de Luca, "A practical method for calculating largest Lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1–2, pp. 117–134, May 1993.

[24] M. E. Cohen, D. L. Hudson, and P. C. Deedwania, "Applying continuous chaotic modeling to cardiac signal analysis," *IEEE Eng. Med. Biol. Mag.*, vol. 15, no. 5, pp. 97–102, Sep./Oct. 1996.

[25] V. L. Schechtman, S. L. Raetz, R. K. Harper, A. Garfinkel, A. J. Wilson, D. P. Southall, and R. M. Harper, "Dynamic analysis of cardiac R-R intervals in normal infants and in infants who subsequently succumbed to the sudden infant death syndrome," *Pediatric Res.*, vol. 31, no. 6, pp. 606–612, Jun. 1992.

[26] I. A. Rezek and S. J. Roberts, "Stochastic Complexity Measures for Physiological Signal Analysis," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 9, pp. 1186–1191, Sep. 1998.

[27] A. Porta, G. Baselli, D. Liberati, N. Montano, C. Cogliati, T. Gnecchi-Ruscone, A. Malliani, and S. Cerutti, "Measuring regularity by means of a corrected conditional entropy in sympathetic outflow," *Biol. Cybern.*, vol. 78, no. 1, pp. 71–78, 1998.

[28] S. M. Pincus and A. L. Goldberger, "Physiological time-series analysis: what does regularity quantify?" *Am. J. Physiol.*, vol. 266, no. 4, (*Heart Circ. Physiol.*, vol. 35), pp. H1643–H1656, Apr. 1994.

[29] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Am. J. Physiol. (Heart Circ. Physiol.)*, vol. 278, no. 6, pp. 2039–2049, Jun 2000.

[30] M. Costa, A. L. Goldberger, and C.-K. Peng, "Multiscale Entropy Analysis of Complex Physiologic Time Series," *Phys. Rev. Lett.*, vol. 89, no. 6, p. 068102, Aug. 2002.

[31] K. Waheed and F. M. Salam, "A Data-Derived Quadratic Independence Measure for Adaptive Blind Source Recovery in Practical Applications," in: *Proc. 45th IEEE Int. Midwest Symposium on Circuits and Systems*, 2002, pp. 473–476.

[32] H.-B. Xie, J.-Y. Guo, and Y.-P. Zheng, "Fuzzy Approximate Entropy Analysis of Chaotic and Natural Complex Systems: Detecting Muscle Fatigue Using Electromyography Signals", *Ann. Biomed. Eng.*, vol. 38, no. 4, pp. 1483–1496, Apr. 2010.

[33] F. Jovic, D. Krmpotic, and A. Jovic, "Process entropy and informational macrodynamics in a ceramic tile plant," in *Proc. 32nd Croatian Society for Information and Communication Technology, Electronics and Microelectronics - MIPRO*, 2009, vol. III, pp. 50–53.

[34] M. Costa, A. L. Goldberger, and C.-K. Peng, "Broken asymmetry of the human heartbeat: Loss of time irreversibility in aging and disease," *Phys. Rev. Lett.*, vol. 95, no. 19, p. 198102, Nov. 2005.

[35] X.-S. Zhang, Y.-S. Zhu, N. V. Thakor, and Z.-Z. Wang, "Detecting Ventricular tachycardia and fibrillation by complexity measure," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 5, pp. 548–555, May 1999.

[36] C. Kamath, "A new approach to detect congestive heart failure using sequential spectrum of electrocardiogram signals," *Med. Eng. & Phys.* vol. 34, no. 10, pp. 1503– 1509, Dec. 2012.

[37] R. U. Acharya, K. P. Joseph, N. Kannathal, C. M. Lim, and J. S. Suri, "Heart rate variability: a review," *Med. Biol. Eng. Comput.*, vol. 44, no. 12, pp. 1031–1051, Dec. 2006.

[38] C.-K. Peng, S. Havlin, H. E. Stanley, and A. L. Goldberger, "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series," *Chaos*, *Solitons, & Fractals*, vol. 5, no. 1, pp. 82–87, Jan 1995.

[39] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proc. of the Royal Society A*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[40] J. P. Zbilut, N. Thomasson, and C. L. Webber, "Recurrence quantification analysis as a tool for nonlinear exploration of nonstationary cardiac signals," *Med. Eng. & Phys.*, vol. 24, no. 1, pp. 53–60, Jan. 2002.

[41] T. B. Garcia and N. E. Holtz, *12-Lead ECG: The Art of Interpretation*, Sudbury, MA, USA: Jones and Bartlett Publishers, 2001.

[42] F. S. Leite, A. F. da Rocha, and J. L. A. Carvalho, "Matlab software for detrended fluctuation analysis of heart rate variability," in *Proc. 3rd Int. Conf. Bio-inspired Systems and Signal Processing BIOSIGNALS*, 2010, pp. 225–229.

[43] P. Perakakis, M. Joffily, M. Taylor, P. Guerra, and J. Vila, "KARDIA: A Matlab software for the analysis of cardiac interbeat intervals," *Comp. Meth. Prog. Biomed.*, vol. 98, no. 1, pp. 83–89, Apr. 2010.

[44] A. Schlögl, C. Vidaurre, E. Hofer, T. Wiener, C. Brunner, R. Scherer, and F. Chiarugi, "Biosig – standardization and quality control in biomedical signal processing using the biosig project," in *Proc. 1st Int. Joint Conf. Biomedical Engineering Systems and Technologies BIOSTEC*, 2008, pp. 403–409.

[45] J.-P. Niskanen, M. P. Tarvainen, P. O. Ranta-aho, and P. A. Karjalainen, "Software for advanced HRV analysis," *Comp. Meth. Prog. Biomed.*, vol. 76, no. 1, 73–81, Oct. 2004.

[46] L. Rodriguez-Linares, X. Vila, A. Mendez, M. Lado, and D. Olivieri, "RHRV: An R-based software package for heart rate variability analysis of ECG recordings," in *Proc. 3rd Iberian Conference in Systems and Information Technologies (CISTI)*, 2008, pp. 565–574.

[47] J. E. Mietus and A. L. Goldberger. (2012) Heart Rate Variability Analysis with the HRV Toolkit. [Online]. Available: http://physionet.org/tutorials/hrv-toolkit/

[48] D. Kaplan and P. Stafin. (2012) Software for Heart Rate Variability. [Online]. Available: http://www.macalester.edu/~kaplan/hrv/doc/