ISSN NO. 2320-5407

Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/7300
DOI URL: http://dx.doi.org/10.21474/IJAR01/7300

RESEARCH ARTICLE

APPLICATION OF INSTANCE-BASED LEARNERS FOR ARRHYTHMIA DETECTION IN ECG SIGNALS.

Andrius gudiškis.
Department Of Electronic Systems, Vilnius Gediminas Technical University.

Manuscript Info

Manuscript History

Received: 16 April 2018
Final Accepted: 18 May 2018
Published: June 2018

Abstract

Cardiac arrhythmia detection in heart activity related signals, like ECG, has been a center of attention for many researchers in medical field, and is as relevant, if not more, to this day. Aim of this paper is to investigate the accuracy of commonly used classifiers in classifying ECG signal segments of one heart beat into three classes – normal, unrecognizable and arrhythmia – and provide insight, which type of, or even specific classifier has a tendency to perform the best under given scenario. Results of the experiments showed that best performing classifiers are instance-based learning algorithms, top two performers being K* algorithm, based on entropic distance measure, with 99% correlation and 3,5% relative absolute error, while testing all input data as test data, 90% and 20% respectively, when testing with 75% of input data as training set, and the rest as testing set, along with IBk – nearest neighbor based algorithm, which was only applicable with percentage split training method (75% / 25%), resulting in 84% correlation and 19% relative absolute error.

Introduction:-

Analysis of heart related activity has been one of the main areas of focus for medical experts and engineers alike for decades now. From a point of engineering, there is already a lot of progress done in designing automated solutions for various parameters of heart activity. Once the age of personal computers has dawned, which marked a significant increase in computational power, many solutions for real-time processing of the most descriptive signal for heart activity – electrocardiogram (ECG) – started to surface. A perfect example of that is the Pan-Tompkins algorithm for real-time detection of QRS complexes in the ECG signal, proposed by J. Pan and WJ. Tompkins in 1985 (Pan et al. 1985), which is to this day is considered to be the most common algorithm for QRS detection in various types of monitors in intensive care units (or ICUs), ambulances and other hardware which requires real-time processing of ECG signal.

While QRS detection in digital form has been introduced back in 1980's and, at least at first, was used in deriving the most common parameters of heart activity like heart rate and rhythm in digital form which are mostly a matter of convenience for the medical personnel, to help them conclude the diagnosis more efficiently and accurately. But during a past decade or so QRS digital detection has sparked considerably high amount of research for using the computers not only for everyday regular tasks is medical wards, but for detecting more crucial and vital information, namely – cardiac arrhythmias.

Corresponding Author:- Andrius gudiškis.
Address:- Department Of Electronic Systems, Vilnius Gediminas Technical University.

Some researcher and engineers focus on single arrhythmia detection (Ortigosa et al. 2015; Felix et al. 2015; Chandra et al. 2016), others aim to detect arrhythmias in general, mostly for two reasons – either to initiate alarms to alert the medical personnel or to reduce the mentioned alarms that might have been detected incorrectly in order to reduce stress on medical personnel, utilizing either own built algorithms (Eerikäinen et al. 2015) or machine learning (Daluwatte et al. 2015). Machine learning applications are becoming very popular for solving these kind of problems, because of their constantly increasing performance and accuracy, and most importantly – ability to be applied to very wide range of different problems.

Since machine learning is gaining more interest by year and evolving equally fast, in this paper the most well-known methods of machine learning are investigated, such as neural networks, decision trees and others to see which one performs the best in creating models for classification of different ECG signal features into classes of normal, distorted by noise and cardiac arrhythmic-signal. Software called WEKA was chosen in order to perform experiments with classifiers – open source software suite for machine learning, which utilizes an impressive range of different classification methods.

Mentioned software was used to measure the accuracy of all available classifiers, in order to conclude which is performing the best when trying to tackle the problem of classifying between normal, noisy and arrhythmic heart activity. Lastly results are discussed and further possible investigations are considered on how the performance could be improved even more.

Materials and methods:-

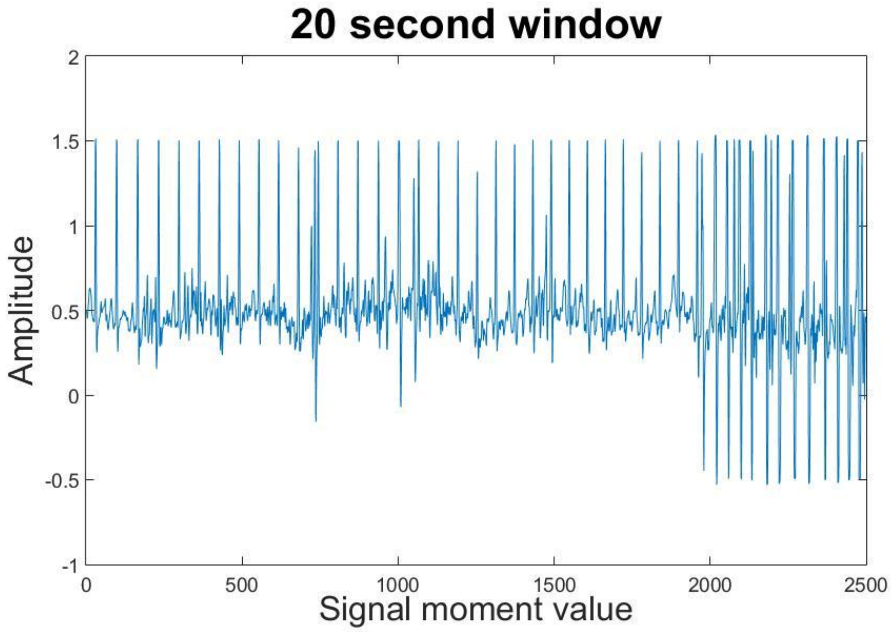


Fig. 1:-20 second wide signal window of one patient, selected for further experiments

Experiments, described in this paper, were performed using PhysioBank MIMIC-III database by Physionet.org. Signals in the mentioned database varies in length from 5 minutes to in some case even an hour long measurements, but considering objective of this research investigation scope was limited to 20 seconds of individual signal measurements for one patient – higher emphasis was put on having more varied signals, with different shape of QRS, different forms of cardiac arrhythmias and different magnitudes of noise instead on analyzing large amounts of data for one signal.

Mentioned 20 second windows of individual signals were not chosen randomly, they are manually picked segments where strong deformation in otherwise normal ECG waveform is clearly noticeable (Fig. 1.). These 20 seconds of ECG signals are then divided in single heart beat (QRS) segments by taking signal annotations, provided of expert physicians (annotations are also available in PhysioBank database), and centering a 100 sample wide window on it

(Fig. 2) – signals are sampled by 125 Hz frequency, so mentioned windows is about the width of one normal heart beat.

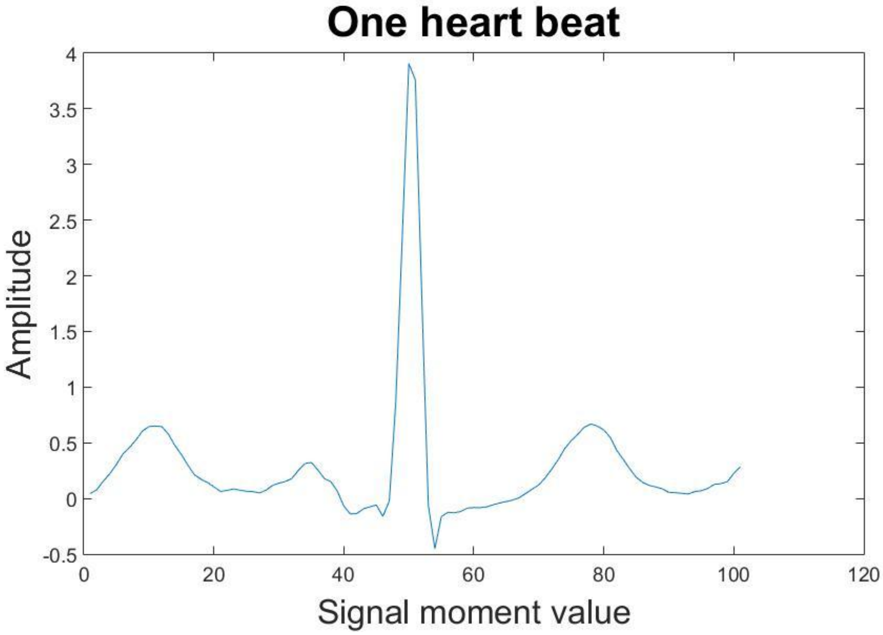


Fig. 2:-100 sample window – one heart beat (QRS)

To make the problem of classification easier, several features of the signal are extracted for each individual QRS – mean value, standard deviation, variance, covariance, kurtosis and sum of squared difference of signal sample amplitude. It is possible to feed all sample values of each QRS window sample as input to the classifier, but that is considered to be a very primate approach and more prone to error, when compared with feature based classification, if the latter one is done right and well performing features are picked.

Each feature record for single QRS window is then assigned a signal state identifier in order to perform a supervised training method for our classifier. As was mentioned before, 100 sample windows were assigned with one of three indicators of signal state: 1 – normal heart beat (Fig.2); 2 – unrecognizable/distorted by noise (Fig. 3a); 3 – cardiac arrhythmia (Fig. 3b).

ECG signals of 40 different patients were preprocessed as described above, which resulted in experimental dataset with states, presented in Table 1.

Described experimental data was used as input to the list of classifiers, provided by the software suite WEKA. 28 different classifiers were tested in total. Training set and percentage split training methods were used. Parameters for each classifier were kept as defaults recommended by WEKA, as the objective of the experiment is to measure general performance of all available methods of classification.

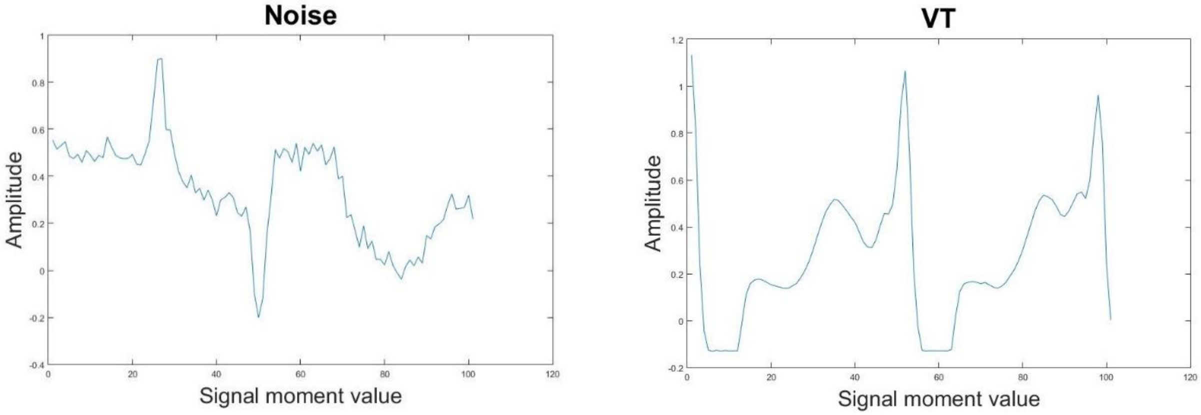
Table 1:-Experimental dataset states

State identifier name	**State identifier**	**Number of records**
Normal heart beat	1	414
Unrecognizable/noise	2	230
Cardiac arrhythmia	3	609

Classifiers used are divided into six categories:

1. **Function-type classifiers** – Gaussian Processes, Linear Regression, Multilayer Perceptron, Simplified Linear Regression, SMO Algorithm for SVM Regression (SMOreg).
2. **Lazy-type classifiers –** Nearest Neighbors (IBk) (Aha et al. 1991), K* (KStar) (Cleary et al. 1995), Locally Weighted Learning (LWL) (Frank et al. 2002).

3. **Meta-type classifiers** – Additive Regression, Bagging, Cross Validation Parameter Selection, Multi-Scheme, Random Committee, Randomized Filter, Random Sub-space, Regression by Discretization, Stacking, Voting, Weighted Instances Handler.
4. **Rule-type classifiers –** Decision table, M5 Rules, ZeroR.
5. **Decision tree-type classifiers –** Decision Stump, M5Base tree, Random Forest, Random Tree, REPTree.



a b

Fig. 3:-Examples of two types of abnormal signal states used in experiments. **a** – unrecognizable/distorted heartbeat, **b** – arrhythmic heartbeat

Results:-

For classifier performance evaluation five parameters were taken into consideration – correlation coefficient, mean absolute error, root mean squared error, relative absolute error and root relative squared error. Results for using whole data as training set are described in Table 2. Some classifiers returned with correlation coefficient as zero (and consequentially relative absolute error as 100%), indicating that classifier modeled values has no correlation to real values. This does not mean that said classifiers are bad, they are just not fit solving problems with either the data format we've provided as input or training method is not one of our picked methods for this investigation – we will not be considering these classifiers in our further measurements.

Table 2:-Classifier results after using whole input data as training set

Classifier	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error (%)	Root relative squared error (%)
GaussianProcesses	0.6899	0.5159	0.6446	62.834	72.4009
LinearRegression	0.6894	0.5148	0.6449	62.7031	72.4368
MultilayerPerceptron	0.7795	0.4192	0.6117	51.0584	68.7054
SimpleLinearRegression	0.6866	0.6866	0.6473	63.1868	72.7032
SMOreg	0.6882	0.478	0.6666	58.2196	74.8754
Ibk	1.0000	0.0000	0.0000	0.0000	0.0000
KStar	**0.9945**	**0.0294**	**0.0949**	**3.5775**	**10.6546**
LWL	0.6925	0.5165	0.6423	62.9092	72.1457
AdditiveRegression	0.8065	0.3781	0.5264	46.0565	59.1215
Bagging	0.9221	0.2003	0.3478	24.3929	39.067
CVParameterSelection	0.0000	0.821	0.8903	100.0000	100.0000
MultiScheme	0.0000	0.821	0.8903	100.0000	100.0000
RandomCommittee	1.0000	0.0000	0.0000	0.0000	0.0000
RandomizableFilteredClassifier	1.0000	0.0000	0.0000	0.0000	0.0000
RandomSubSpace	0.9129	0.2825	0.3851	34.4089	43.2576
RegressionByDiscretization	0.9475	0.101	0.2847	12.3041	31.9739
Stacking	0.0000	0.821	0.8903	100.0000	100.0000

Vote	0.0000	0.821	0.8903	100.0000	100.0000
WeightedInstancesHandlerWrapper	0.0000	0.821	0.8903	100.0000	100.0000
InputMappedClassifier	0.0000	0.821	0.8903	100.0000	100.0000
DecisionTable	0.8108	0.3305	0.5211	40.2557	58.5314
M5Rules	0.8928	0.2602	0.4041	31.6907	45.3869
ZeroR	0.0000	0.821	0.8903	100.0000	100.0000
DecisionStump	0.6879	0.5223	0.6462	63.6135	72.5829
M5P	0.8899	0.2604	0.4092	31.7178	45.9645
RandomForest	0.9876	0.0764	0.1471	9.3118	16.519
RandomTree	0.0000	0.0000	0.0000	0.0000	0.0000
REPTree	0.8595	0.2547	0.455	31.0287	51.1074

Instance-based learning algorithm K* performs the best with proposed input data while using all of the input data as a training set, displaying correlation close to 100% (99.45%) and relative absolute error 3.5%. Second and third best classifiers for the first part of the experiment are Regression by Discretization and Random Forest algorithms with 98.76% and 94.75% correlation respectively. It is also worth noting that there are a few instances where certain classifiers have resulted in correlation of 100% and error 0%. Sadly, that does not indicate perfect choice for given problem solving, quite the opposite – it shows that these classifiers are not testing themselves against the training set, and need separate set of data for testing the classification model. As a result, output of mentioned classifiers is more realistic looking, when performing second part of the experiment – training the classifier with 75% of data and using the rest as testing set. Also a drop in accuracy for all classifiers can be witnessed, but that is to be expected, since algorithms are working with less data while training, which lets us safely assume that the more date of same type classifier gets for training procedure, the more accurate the created model is.

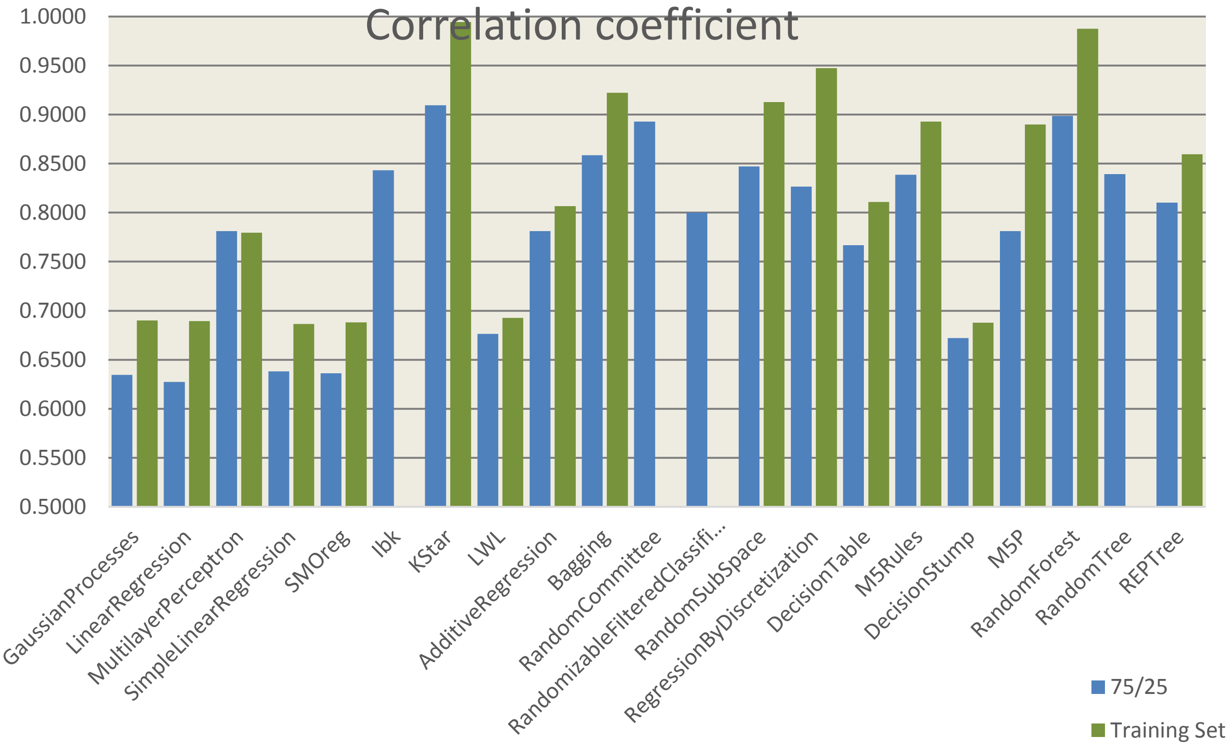


Fig. 4:-Correlation result of all classifiers

Table 3:-Classifier results after 75% of the input data as training set, and the rest for testing (not applicable classifiers excluded)

Classifier	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error (%)	Root relative squared error (%)
GaussianProcesses	0.6345	0.5547	0.6961	66.5907	77.6915
LinearRegression	0.6274	0.5506	0.703	66.0946	78.4558
MultilayerPerceptron	0.7811	0.7351	0.8656	88.2453	96.6031
SimpleLinearRegression	0.6381	0.5512	0.6928	66.1701	77.3189
SMOreg	0.6362	0.5118	0.7128	61.4453	79.5546
Ibk	0.8433	**0.1597**	0.4992	**19.1772**	55.7137
KStar	**0.9095**	0.1706	**0.375**	20.4785	**41.8551**
LWL	0.6764	0.5277	0.6595	63.3522	73.5991
AdditiveRegression	0.7813	0.4039	0.5592	48.4857	62.4147
Bagging	0.8585	0.2863	0.4606	34.3712	51.4088
RandomCommittee	0.8928	0.2032	0.4039	24.3934	45.077
RandomizableFilteredClassifier	0.8002	0.2077	0.5624	24.9303	62.7671
RandomSubSpace	0.8471	0.3965	0.5012	47.5945	55.9326
RegressionByDiscretization	0.8264	0.2258	0.511	27.1131	57.0292
DecisionTable	0.7669	0.3919	0.5747	47.0454	64.141
M5Rules	0.8386	0.3085	0.4876	37.0333	54.4222
DecisionStump	0.672	0.533	0.6633	63.988	74.0233
M5P	0.7813	0.3313	0.5641	39.7774	62.9606
RandomForest	0.8988	0.2255	0.3942	27.0743	43.9999
RandomTree	0.8392	0.1789	0.5056	21.4784	56.4235
REPTree	0.8103	0.3075	0.5259	36.9201	58.6901

Another notable point is that during the second part of the experiment the top performing algorithm remained instance-based learner K*, even though it's relative absolute error increased to 20% and correlation decreased by 9% (Fig. 4, 5). Second best, considering its relatively high correlation coefficient (0.8433) and lowest relative absolute error – 19%, that makes two of the best performing algorithms instance-based learners. Decision tree algorithms have performed considerably well in the second part of the experiment too, specifically – Random Tree and Random Forest algorithms which performed just as effectively as previously mentioned instance-based learners.

Conclusions and discussion:-

While experimenting with preprocessed data of ECG signal in a form of feature set for single QRS complexes in order to classify the mentioned records into three classes – normal, distorted by noise or unrecognizable and cardiac arrhythmia – 28 different classifiers were tested using two training methods – all input data as training set and percentage split – 75% as training set and 25% as testing set. During tests with the first training method it quickly became apparent that instance-based learning algorithm K* performed the best out of the rest, displaying the highest correlation between predicated and actual values and lowest relative absolute error.

In second phase of the experiment where percentage split K* again displayed very promising performance and stood out of the rest of the classifiers, this time along with another instance-based learning algorithm – IBk. This made it safe to conclude, that instance-based learning algorithms are a solid choice for this specific classification scenario, followed by decision tree based algorithms, which were relatively very close in accuracy to the mentioned K* and IBk.

Even though the objective of this experiment was to measure general performance or accuracy of large set of classifiers, there is still room for improvement in both data preparation and preprocessing as well as algorithm tweaks:

1. During the performed test the tendency was noticed, that the performance of most classifiers directly relates to the amount of training data provided. The more data we can feed to the input of the algorithm the more likely it is to perform better. In this experiment we've used around 1250 unique records, all of them were used for

training in the first part of the experiment and 75% of them – in the second part. But that is still relatively small amount of data when considering machine learning. These algorithms are capable to process 10 or even 100 times more input data which would most likely positively affect the overall accuracy of the model.

2. Our approach to the classification might have been too straightforward and could've made classification problem more complex than it could've been otherwise. As mentioned, three distinct signal states were used as classes in our experiments – normal, noisy and arrhythmic heartbeat. It is possible that accuracy would improve if we would do two classification models instead of one – by classifying the data into normal and abnormal first, and classifying the abnormal records into unrecognizable or arrhythmias afterwards. This theory will be investigated in future experiments.
3. As the aim of the work was to measure general performance of investigated classifiers, recommended default settings by WEKA were used for all classifiers in the experiments. After choosing the best performing classifiers it is possible that their accuracy could be further improved by tweaking the mentioned settings.
4. More features could be introduced in the data preprocessing stage, what would make the input data potentially more informative for the classifier, making its problem less complicated.

References:-

1. Pan, J. and Tompkins, W.J., 1985. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering*, (3), pp.230-236.
2. Ortigosa, N., Cano, Ó., Galbis, A. and Fernández, C., 2015, September. Classification of atrial fibrillation episodes by means of phase variations of time-frequency transforms. In *Computing in Cardiology Conference (CinC)*, 2015 (pp. 41-44). IEEE.
3. Felix, J., Alcaraz, R. and Rieta, J.J., 2015, September. Adaptive wavelets applied to automatic local activationwave detection in fractionated atrial electrograms of atrial fibrillation. In *Computing in Cardiology Conference (CinC)*, 2015 (pp. 45-48). IEEE.
4. Chandra, B.S., Sastry, C.S. and Jana, S., 2016, September. Subject-specific detection of ventricular tachycardia using convolutional neural networks. In *Computing in Cardiology Conference (CinC)*, 2016 (pp. 53-56). IEEE.
5. Eerikäinen, L.M., Vanschoren, J., Rooijakkers, M.J., Vullings, R. and Aarts, R.M., 2015, September. Decreasing the false alarm rate of arrhythmias in intensive care using a machine learning approach. In *Computing in Cardiology Conference (CinC)*, 2015 (pp. 293-296). IEEE.
6. Daluwatte, C., Johannesen, L., Vicente, J., Scully, C.G., Galeotti, L. and Strauss, D.G., 2015, September. Heartbeat fusion algorithm to reduce false alarms for arrhythmias. In *Computing in Cardiology Conference (CinC)*, 2015 (pp. 745-748). IEEE.
7. Aha, D.W., Kibler, D. and Albert, M.K., 1991. Instance-based learning algorithms. *Machine learning*, 6(1), pp.37-66.
8. Cleary, J.G. and Trigg, L.E., 1995. K*: An instance-based learner using an entropic distance measure. In *Machine Learning Proceedings 1995* (pp. 108-114).
9. Frank, E., Hall, M. and Pfahringer, B., 2002, August. Locally weighted naive bayes. In Proceedings of the *Nineteenth conference on Uncertainty in Artificial Intelligence* (pp. 249-256). Morgan Kaufmann Publishers Inc.