An E-Health System for Data Stream Analysis

Diego Alysson Braga Moreira, Levy Gurgel Chaves,
Rafael Lopes Gomes and Celestino Júnior Joaquim

July 28, 2020

An E-Health system for data stream analysis

Diego A. B. Moreira, Levy G. Chaves, Rafael L. Gomes, Joaquim Celestino Jr.
State University of Ceara (UECE) Fortaleza – CE, Brazil
Email: {diego,levy,rafaelllgom,celestino}@larces.uece.br

Abstract—**E-Health technologies arose as a suitable approach to support diseases diagnostics and treatment decisions since the Internet of Things devices can monitor humans over a long period. Most of the E-Health technologies are based on machine learning to analyze and classify patients data, returning a possible diagnosis for health professionals as fast and accurately as possible. However, machine learning techniques have high computational complexity, limiting their usage to meet the real time requirements of E-Health systems. Within this context, this paper proposes an E-Health system to analyze and to classify diseases in patients with biological exams on data streams, using machine learning techniques. The applied data stream approach enables the online training of the classifiers, as well as a suitable performance for data processing. The experiments performed were based on a database of real patients. The results (considering 19 different anomalies) suggest the feasibility of the proposed E-Health system, reaching 96%, 94.21%, 92.14%, and 92.53% of accuracy, precision, sensitivity, and cover index, respectively, overcoming the existing solutions.**

Index Terms—**Internet of Things, E-Health, Data Stream, Machine Learning.**

I. INTRODUCTION

In the last few years, the smart device industry has expanded, following the device market and the ease of smart objects production. These types of equipment are used in several market areas, from industry to specific applications in smaller scenarios. Devices that previously required a large appliance now can be deployed on small hardware, such as wearable and portable devices.

One possible scenario is the application of these devices for medicine, due to the great concern with healthy habits and continuous monitoring of patients. This scenario is called E-Health. Previously, E-Health technology was used occasionally through small medical interventions or video calls. Currently, the e-Health systems are promoted by the advances of embedded technologies and society's greater concern for health.

The medical devices, together with today's networked structures, are part of the so-called Internet of Things (IoT). With the evolution of this new paradigm, the use of a wide range of devices is being considered, collecting information, and acting in the environments. Regarding medical applications, these IoT devices collect information about the patients and generate diagnosis alerts. As a consequence, IoT devices are used to monitor patients continuously, supporting disease diagnostics and observation tools. In this continuous monitoring scenarios, all data must be collected and then entirely analyzed, allowing an acceptable response of medical applications.

One of the most important applications of E-Health systems is the diagnosis of diseases, i.e., the E-Health systems support health professionals to identify/classify complex human diseases that need a high level of technical training. The main approach to be applied in the E-Health systems is the machine learning techniques, that classify possible anomalies in medical exams based on knowledge from previous data analysis. Several machine learn models were proposed for the classification of different patterns in the collected data set, such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks.

This constant monitoring of a large data flow generation represents scenarios with a large number of patients. All of this data needs to be used for classification and prediction of new diseases, but there is great difficulty in storing, classifying, and especially training the model through this large amount of data.

A suitable approach to improve the performance of data classification (from machine learning perspective) is the partitioning of classification into periods though data streams (or flows). The idea of this approach is to collect data in a specific period related to a singular patient and train the classifier to attach knowledge to a previously trained classifier. In this way, this updated classifier can perform the patient diagnoses efficiently.

The data stream approach assumes that the amount of data and frequency generation are high, where this data need to be analyzed and results extracted as fast and accurate as possible. For example, in a hospital environment, patients need to be monitored frequently and their clinical condition may change quickly. Thus, it necessary a model to monitor and track patients' progress considering their individual and past history, as well as the usual behavior of the diseases.

Within this context, this paper proposes an E-Health system to analyze and to classify patients' data based on data streams. The proposed data stream model processes the received data from a specific monitored patient and then submits the processed data to the machine learning engine to update its knowledge according to the current state of the monitored patient. This approach enables an intelligent disease classification with runtime efficiency and high accuracy.

The experiments performed used MIT-BIH database of real patients [1], which is widely used in the literature. This fact allowed the performance comparison of the proposed E-Health system with existing solutions (such as [2]–[6]). The results suggest the feasibility of the proposed E-Health system, overcoming the existing solutions.

The remainder of this paper is organized as follows: Section II presents the existing related work. Section III introduces

TABLE I
RELATED WORK.

Paper	**Method**	**Number of Anomalies**	**Computational Cost**	**Adaptability to new diseases**	**Number of Attributes**
Acharya et al. [7]	Wavelets + ECG Segmentation + CNN	4	High	No	4
Tang et al. [8]	QRS Detection + Filters + SVM	5	Low	No	5
Bensujin et al. [9]	K-means + Genetic Algorithms	2	High	No	2
Tavassoli et al. [10]	ECG and HRV Features + Genetic Algorithms	7	Medium	No	15
Martis et al. [11]	Waletet + ICA + KNN	3	High	No	5
This Work	Data preprocessing + Interest Blocks + Leveraging Bagging (Online Methods)	18	Low	Yes	10

the proposed E-Health system for data stream analysis, while Section IV describes the results of the experiments performed. Finally, Section V concludes the paper and presents future work.

II. RELATED WORKS

The main related works will be presented in this session, considering data collection and detection of anomalies in biological environments to the employment of learning machines to classify these anomalies. We will also evaluate approaches that use flow learning algorithms.

Acharya *et al.* [7] propose a two-step approach for classifying ECG signals. The first processing step consists of removing all noise from the signal and the baseline is removed by Daubechies wavelets. Subsequently, the ECG signal is segmented and delivered as input to a neural network. The Z-score normalization is applied to each segment and the proposal uses two neural networks for classification. These networks are 11-layer Convolutional Neural Network (CNN). Kernel sizes of 27 were used for convolution, the activation functions used were leaky rectified linear unit and Softmax. Based on an incremental approach, this type of approach requires a huge amount of training data and the training step requires a long time, and this type of method is not in line with the criterion of adaptation in iterative scenarios.

Tang *et al.* [8] classify the ECG signal by extracting features from the time domain and the frequency domain. These features are then used as input to *Support Vector machines*(SVM). SVM can be defined as a tool widely used in binary problems because it has good classification performance. In summary, SVM seeks to find the maximum margin between the training data and the decision boundary. In a given phase, the method performs a mapping between feature vectors in a larger dimension space and this can become a problem for the ECG case, as for spaces having high dimensionality SVM will also have a high error rate [12].

In the method proposed by Bisujin [9], they decided only to examine the ST-Segment Elevation Myocardial Infarction (STEMI) in the ECG signal. A hybrid detection approach was used to determine cardiac rates, ischemic, and extraction of STEMI. The hybrid approach consisted of using the K-means algorithm to perform feature extraction and Bacterial Foraging Optimization Algorithm (BFOA) is used to detect STEMI in ECG signals. Despite the good results, the preprocessing step proved inefficient.

Tavassoli *et al.* [10] propose a 4-phase online system for feature extraction and detection of cardiac arrhythmias based on a clustering methodology. The system is divided into two parts, an online and offline for data processing. The offline step adopts a system of inferences rules obtained from various information about patients, such as gender, symptoms, age, among others. In the online stage, the ECG is received as input and it suffers from initial preprocessing, to remove noise from the signal acquisition process. After that, time-domain features are extracted from the wave, for example, RR interval, root mean square successive difference (RMSSD) in 8 RR intervals (in heart period series is a time-domain measure of heart period variability), fractal dimension, Hurst exponent, etc. The extracted features are then sent to a data mining algorithm. Finally, the algorithm will form clusters of similar data, and next to the offline rules will be decided on the type of arrhythmia.

Martis *et al.* [11] proposed a method for classifying atrial arrhythmias. To this end, it implemented a simplified version of the Pan-Tompkins [13] algorithm, which aims to identify the full QRS. After the defined blocks, just over 200 features are extracted from the ECG signal using 3 order spectra (HOS) coefficients are calculated in the processing. To perform the analysis of all features is used the PCA method to reduce the dimensionality of the problem to 12. In the classification stage, three supervised algorithms were states, namely: k-means, Fuzzy c-means, and expectation-maximization clustering. The proposal was validated at MIT-BH and the European ST-T ischemia database.

III. PROPOSAL

In this section, we present the purpose of this paper to classify biological anomalies in a data flow environment. For this, we will divide the section into three main parts, which are: data acquisition and preprocessing, an overview of the learning machines used, and lastly, the proposed training of machines through data flow environment.

A. Data Acquisition and Preprocessing

Currently, several E-Health devices can perform a constant collection of patient biological data. For this, there must first be a network capable of receives this data and forwards it to the learning modules, which must then be trained and classified from this type of information.

An IoT framework is proposed, where E-Health devices can send some information to a central receiver. This device will be in charge of collecting the information being measured by the sensors and first preprocessing the data.

This processing step is built by two main steps. Initially, the data is split according to the type of signal received. Each type of data needs different treatments to be used as learning material. Discrete data such as temperature, pressure, oxygenation, are separated and analyzed according to the space of occurrence and collection so that repeated signals or errors in the collection should be discarded. In this phase, errors are detected by analyzing the patterns of each data. It is mainly analyzed the biological possibility of the occurrence of each type of the collected signals.

Another type of data accepted, of continuous collection, such as ECG (Electrocardiogram) and EEG (Electroencephalography) waves must go through processes of data filtering and collection of useful information, extracting points of importance for each one of these signals, as performed in [14] that presents a process of filtering of undesired artifacts in ECG waves, utilizing filters based on Discrete Wavelet Transforms and performs a process of information extraction using techniques of moving averages.

After performing the preprocessing steps, it is necessary to prepare the data for use as input to the machine learning algorithms. For this, a characteristic block scheme is proposed. These blocks are formed by parts of the patient's biological signals. A portion of data of interest is defined as the smallest portion that a biological signal can represent a patient's current state.

Thus, the data of a particular patient is represented by a a set of blocks that represent the patient's current health. These blocks thus store all the essential characteristics of the patient, which will be used for training of the learning machines and classification. As an visual example of a block set, observed at the image 1, this block has discrete and non-discrete information such as ECG, temperature, respiratory rate, blood pressure, heart rate, and oxygen saturation.

After the preprocessing, with all the data prepared and the patients represented as a set of characteristic blocks, it is desired to pass this data as input to the next phase of learning machine training. For this, different algorithms and configurations were used.

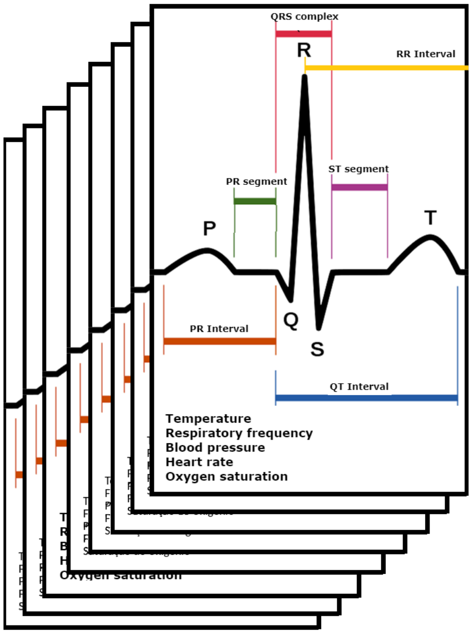

Fig. 1. Example of feature block sets.

B. Online Machine Learning

The need for in-stream data analysis processing was quickly encouraged by large demands from industries, financial transactions, and other applications. Some points differ from conventional machine learning from online learning, based on three main requirements: 1) model upgrade after each new instance received without retraining with previous instances; 2) adaptive models are capable of dealing with concept drift; 3) Online learning methods need to be ready to be available in distributed environments, due to the large volume of data it is impracticable to leave the classification centrally.

Concept drift refers to the change in the data distribution from its order of arrival, that is, at a given moment t the input data follows another probability distribution pattern. This distribution may be given gradually over time or abruptly. For this, our model needs a mechanism to detect these changes and make the necessary treatment for readaptation taking into account the new distribution of samples. One of the methods designed to deal with this concept is ADWIN [15].

The main idea behind ADWIN is to keep statistics when detecting concept drift from a window of variable length. It keeps a variable-length window of recently seen items, with the property that the window has the maximal length statistically consistent with the hypothesis "There has been no change in the average value inside the window". More precisely, an old fragment of the window is dropped if and only if there is enough evidence that its average value differs from that of the rest of the window. This has two consequences: one, change is reliably detected whenever the window shrinks; and two, at any time the average over the current window can be used as a reliable estimate of the current average in the stream.

As a result, stream learners need to build models that evolve. Processing will depend only on the order of examples generated from a given continuous data stream whose distribution of the data may be changed. One of its great advantages is the ability to be available to perform assessments and make predictions at any time. However, the area presents some challenges, such as online parameter adaptation, concept drift, a model adaptation based on a limited number of samples, and limitations in memory usage.

Algorithms selected for training and classifying anomalies in a real-time data flow scenario must have the ability to perform constant training as new observations are collected, machine learning must adapt to new knowledge, forming thus a solid base and representing the anomalies. For this, Hoeffding Tree, Hoeffding Anytime Tree, Leveraging Bagging and Adaptative Random Forest were examined.

1) Hoeffding Anytime Tree: In a data stream environment, where you cannot store all the model's memory, the main problem with building a decision tree is the need to reuse information to compute the best split of attributes. Domingos and Hulten [16] proposed Hoeffding Tree, a fast and efficient classifier for streaming data, where tree construction is based on the arrival of new instances. Its operation is based on statistical methods and *Hoeffding bound*. An important factor is that the tree is built in a way that it tends to converge to the same tree structure built by offline classifiers with a large amount of data. Manapragada et al. [17] use Hoeffding bound to determine whether the merit of splitting on the best attribute except for the merit of not having a split or the merit of the current split attribute.

2) Leveraging Bagging: In the Bagging method, each base classifier is trained in parallel with a training set that is generated by a random override of N examples (or data) from the original training data set, where N is the size of the set of original training, reducing overfitting by averaging or voting, however, this leads to an increase in bias, which is offset by a reduction in variance. Bifet *et al.* [18] propose a new ensemble classification method *Bagging* is proposed. In this way, it was improved by the resampling technique and by using output detections code. Resampling is done using Poisson distribution by assigning different ranges of weights to samples. The key factor of the method is that it effectively trains the classifiers on different subsets of instances without increasing the *bias* of the problem.

3) Adaptative Random Forest (ARF): The ARF [19] method consists of an adaptation of the Random forests method, consisting of Hoeffding trees as a base classifier and has a drift detection agent. The original method of Random forests is a decision tree ensemble, trained using the Bagging method. In ARF, variability is added to the method of performing resampling with Poisson distribution with mean 6. The adaptive part refers to the possibility of adapting the model to concept drift contexts. Another improvement refers to Random forests is that ARF randomly chooses a subset of features per node split. Another is that the detected warning face is created a new background tree, which is trained with the model, but its output disregarded to evaluate the result. Then, when an actual drift is detected by the agent, the worst tree is replaced by the background tree, allowing continuous model improvement without parameter reset. ARF uses the ADWIN adaptive window for warnings and concept drift detection.

The three proposed machine learning will be used both for training and for classifying new entry anomalies. For this, the result obtained in each of the canaries was compared with different methods.

C. Learning Models Training

To classify the blocks of interest, a feature vector was created containing the following inputs: amplitude of points P, Q, R, S, T, duration of intervals PR, PQ, QRS, ST, QT, and RR; following the methodology described in Section III-A). As the proposed work has the purpose of online learning, the models will be trained with a small portion of samples and later the whole learning process will be done online because of the adherence of the classifiers for this purpose. For the models selected in the previous step, they were trained according to the following methodology: 1) each model is initially trained with 150 random blocks of interest samples; 2) It is assumed that only 1 new sample arrives at a time; 3) all new samples will be classified and statistics will be evaluated within a 500 sample window; 4) The evaluation step consisted of evaluating the method using Prequential Evaluation.

For each classifier several subsets of hyperparameters were evaluated. The tables II, III, IV show the tested hyperparameters for each classifier and the best configuration obtained is in bold. N/A indicates not available.

TABLE II
LEVERAGING BAGGING HYPERPARAMETER TESTING

HyperParameter	**Value List**
Base Estimator	**KNN**, Decision Tree, BatchIncremental, OzaBagging
Number of estimators	**17**, 31, 45, 57, 82
ADWIN delta	$\mathbf{10^{-5}}$, $2*10^{-5}, 3*10^{-5}$
Use Code Matrix	**Yes**, No
Poisson Distribution Parameter	5, **6**, 7, 8

TABLE III
ARF HYPERPARAMETER TESTING

HyperParameter	**Value List**
Number of trees	10, 14, 22, **31**, 45, 51
Poisson Distribution Parameter	5, **6**, 7, 8
Drift Detection Method	**ADWIN** [15], EDDM [20]
ADWIN delta	$\mathbf{10^{-5}}$, $2*10^{-5}, 3*10^{-5}$
Grace period	45, 50, **55**, 60, 65
Split criterion	Gini, **Information gain**

TABLE IV
HAT HYPERPARAMETER TESTING

HyperParameter	**Value List**
Grace period	45, 50, **55**, 60, 65
Split criterion	Gini, **Information gain**
min_samples_reevaluate	19, **20**, 22, 25

The Base estimator indicates which is the classifier used in the ensemble method, which is the main classification agent; Number of estimators indicates the number of base estimators to be considered in the ensemble; ADWIN delta is the threshold between the two parse windows; Poisson parameters means

TABLE V
COMPARED METHODS

Classifiers	Accuracy(%)	Precision/Specificity(%)	Sensitivity(%)	Kappa(%)
Hoeffding Tree [21]	88.03	77.13	76.92	74.90
Hoeffding Anytime Tree [17]	92.81	92.15	89.99	84.95
Adaptative Random Forest [19]	94.53	91.87	90.68	87.52
Leveraging Bagging [18]	96.33	94.21	92.14	92.53

TABLE VI
COMPARISON WITH EXISTING SOLUTIONS

Paper	Accuracy(%)	Precision(%)	Recall(%)	Nº of Anomalies
Nurmaine et al. [2]	92.40	N/A	89.00	9
Li et al. [3]	94.61	N/A	N/A	4
Zhang et al. [4]	90.19	92.34	80.35	3
Proposed E-Health System	96.33	94.21	92.14	18

TABLE VII
COMPARISON WITH ONLINE CLASSIFICATION METHODS

Paper	Accuracy(%)	Precision(%)	Recall(%)	Kappa(%)	Nº of Anomalies
Wagnet et al. [5]	94.90	N/A	N/A	N/A	2
Lan et al. [6]	78.895	80.1	78.9	57.8	2
Proposed E-Health System	96.33	94.21	92.14	92.53	18

the average distribution used for resampling with replacement; Number of trees indicates the number of trees used; Drift Detection Method indicates which method will be used for drift detection; number of estimators indicate the number of base estimators to be considered in the ensemble; Grace period informs the number of instances a leaf should observe between split attempts; min_samples_reevaluate indicates the number of instances a tree node must look at to reevaluate the best split; In split criterion is defined what will be the function used for split decisions.

IV. PERFORMANCE EVALUATION

For the proposed qualification method, a disease database with real patients was used. The application was observed relating different arrhythmias in a human ECG, being able to detect 19 different anomalies in this exam. For this, we used the MIT-BIH [1] database of real patients, built at Beth Israel Hospital, and contains 48 outpatient ECG recordings of 30 min each, obtained from 47 different patients. Two or more cardiologists independently noted the records for each heartbeat (approximately 110,000 notes) included in the database, where disagreements were resolved for readability.

Among the 19 anomalies, reported, Normal Heartbeat, Left bundle branch block beat, Right bundle branch block beat, Aberrated atrial premature beat, Premature ventricular contraction, Fusion of ventricular and normal beat, Nodal (junctional) premature beat, Atrial premature contraction, Premature or ectopic supraventricular beat, Ventricular escape beat, Nodal (junctional) escape beat, Paced beat, Left or right bundle branch block, Ventricular flutter wave, Atrial escape beat, Supraventricular escape beat, Non-conducted P-wave (blocked APB) Fusion of paced and normal beat, R-on-T premature ventricular contraction.

Two libraries were used to perform the experiments in a data stream environment, respectively scikit-multiflow [22] and Massive Online Analysis (MOA) [23].

As previously presented, the machines were submitted to the data and were trained inflow, where the new data that was observed formed the learning base, with this it was possible to classify the new samples. The results obtained through the proposed algorithms are presented in table V.

Despite the good results of all methods, Leveraging Bagging was the best among them, with average results of 96.33 % accuracy, 94.21 % accuracy, 92.14 % sensitivity, and 92.53 %. Kappa index. Secondly the Adaptative Random Forest technique. These good results are due to Bagging and Bootstrap techniques, which can reduce the over-fitting of the methods used through multiple learning techniques.

The proposed methods and algorithms used were also compared with other proposals in the literature. For this, two factors were compared mainly, the flow classification capability, compared to traditional classification algorithms, table VI, and a comparison between other flow algorithms VII.

As noted, the technique proposed in this paper, which uses the Leveraging Bagging algorithm, has great results compared to traditional classification algorithms, and has real-time results, with a small amount of data and instant training, giving a great speed of training and classification of the tested models.

Another determining factor of the results obtained is the number of anomalies that each study evaluated, since the larger the quantity, the greater the complexity of the learning base, which can hinder the learning process and classification of anomalies. Nevertheless, the proposed method analyzed 18 different anomalies and presented better results than the others, which have nine, three and two types of anomalies, respectively.

The table VII presents the results obtained compared to two other known works of flow classification. It is observed that the proposed method also presents the best result in this scenario. In addition to these results, an excellent rating is again shown, even with 18 different anomalies, compared to the other methods that analyze only two other anomalies.

V. CONCLUSION

To find a good solution for disease classification in a scenario with constant information flow, this paper proposed an intelligent classification environment, which uses a set of machine learning, capable of training as data comes into the system, then classify this data considering the past and current history of the learning state.

It was first demonstrated in this paper that computer systems are capable of performing biological signal classifications, thus differentiating abnormalities in exams collected from patients. The results obtained, 96 % mean accuracy and 94 % mean sensitivity, demonstrate the high degree of predictability of the proposed system in a data flow environment.

We also analyzed the results obtained compared to traditional learning systems, where the learning base is first set up and well established, and then start the classifications. Through the results of this comparison, it is clear that systems in data streams have a small drop in final classification capacity but can easily adapt to the monitored environment, which qualifies such a proposal with excellent results in this scenario. The proposed stream machine learning is also able to monitor and classify results much earlier than in classic scenarios, as the input data is learned, the next input data can already be classified.

A future approach is expected to use a framework that supports a hybrid scenario where classic learning machines can work in conjunction with flow learning machines, so a higher accuracy of classification is expected in scenarios where each model has better performance. We also want to continue checking different learning models for disease classification in intelligent flow scenarios, optimizing their hyperparameters according to the input anomaly.

REFERENCES

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

[2] S. Nurmaini, R. Umi Partan, W. Caesarendra, T. Dewi, M. Naufal Rahmatullah, A. Darmawahyuni, V. Bhayyu, and F. Firdaus, "An automated ecg beat classification system using deep neural networks with an unsupervised feature extraction technique," *Applied Sciences*, vol. 9, no. 14, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/14/2921

[3] T. Li and M. Zhou, "Ecg classification using wavelet packet entropy and random forests," *Entropy*, vol. 18, no. 8, 2016. [Online]. Available: https://www.mdpi.com/1099-4300/18/8/285

[4] Z. Zhang, J. Dong, X. Luo, K.-S. Choi, and X. Wu, "Heartbeat classification using disease-specific feature selection," *Computers in Biology and Medicine*, vol. 46, pp. 79 – 89, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001048251300348X

[5] T. Wagner, S. Guha, S. Kasiviswanathan, and N. Mishra, "Semi-supervised learning on data streams via temporal label propagation," in *International Conference on Machine Learning*, 2018, pp. 5095–5104.

[6] K. Lan, S. Fong, L.-S. Liu, R. K. Wong, N. Dey, R. C. Millham, and K. K. Wong, "A clustering based variable sub-window approach using particle swarm optimisation for biomedical sensor data monitoring," *Enterprise Information Systems*, vol. 0, no. 0, pp. 1–21, 2019.

[7] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam, "Automated detection of arrhythmias using different intervals of tachycardia ecg segments with convolutional neural network," *Information Sciences*, vol. 405, pp. 81 – 90, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025517306539

[8] X. Tang and M. Zhiwen, "Electrocardiogram classification method based on svm," in *International Conference on Intelligent Systems and Knowledge Engineering 2007*. Atlantis Press, 2007/10. [Online]. Available: https://doi.org/10.2991/iske.2007.282

[9] Bensujin, C. Vijila, and C. Hubert, "Detection of st segment elevation myocardial infarction (stemi) using bacterial foraging optimization technique," *International Journal of Engineering and Technology*, vol. 6, pp. 1212–1223, 05 2014.

[10] Tavassoli, M. Ebadzadeh, and Malek, "Classification of cardiac arrhythmia with respect to ecg and hrv signal by genetic programming," *Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition*, vol. 3, pp. 1–13, 01 2012.

[11] R. J. Martis, U. R. Acharya, H. Prasad, C. K. Chua, C. M. Lim, and J. S. Suri, "Application of higher order statistics for atrial arrhythmia classification," *Biomedical Signal Processing and Control*, vol. 8, no. 6, pp. 888 – 900, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809413001213

[12] S. Wang, "Support vector machines classification for high-dimentional dataset," in *2012 Fourth International Conference on Multimedia Information Networking and Security*, Nov 2012, pp. 315–318.

[13] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, March 1985.

[14] D. A. Moreira, L. G. Chaves, B. A. Lima, K. L. Almeida, T. P. de Araujo, R. L. Gomes, J. Celestino, and A. J. Neto, "Electrocardiogram fiducial points detection for health care systems," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 00 640–00 645.

[15] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," vol. 7, 04 2007.

[16] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*. ACM Press, 2001, pp. 97–106.

[17] C. Manapragada, G. I. Webb, and M. Salehi, "Extremely fast decision tree," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 1953–1962.

[18] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases*, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 135–150.

[19] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9, pp. 1469–1495, Oct 2017. [Online]. Available: https://doi.org/10.1007/s10994-017-5642-8

[20] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, and A. Bifet, "Early drift detection method."

[21] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/347090.347107

[22] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018. [Online]. Available: http://jmlr.org/papers/v19/18-251.html

[23] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?id=1859903