Using Hidden Markov Models and Rule-based Sensor Mediation on Wearable eHealth Devices

Gilles Irénée Fernand Neyens, Denis Zampunieris
University of Luxembourg
Luxembourg
Email:gilles.neyens@uni.lu, denis.zampunieris@uni.lu

***Abstract*—Improvements in sensor miniaturization allow wearable devices to provide more functionality while also being more comfortable for users to wear. The Samsung Simband©, for example, has 6 different sensors Electrocardiogram (ECG), Photoplethysmogram (PPG), Galvanic Skin Response (GSR), Bio-Impedance (Bio-Z), Accelerometer and a thermometer as well as a modular sensor hub to easily add additional ones. This increased number of sensors for wearable devices opens new possibilities for a more precise monitoring of patients by integrating the data from the different sensors. This integration can be influenced by failing or malfunctioning sensors and noise. In this paper, we propose an approach that uses Hidden Markov Models (HMM) in combination with a rule-based engine to mediate among the different sensors' data in order to allow the eHealth system to compute a diagnosis on the basis of the selected reliable sensors. We also show some preliminary results about the accuracy of the first stage of the proposed model.**

***Keywords*–Wearable devices; Conflict handling; Hidden Markov Model; Autonomic Computing; Rule-based Systems; Sensor Mediation.**

I. Introduction

In recent years, advances in sensor technology allow for comfortable wearable devices. This opens new possibilities in different areas as wearable social communities [1] and especially health care [2][3]. A patient can be monitored constantly at home in a non-invasive way, be it during his rehabilitation process [4] or to detect more elusive conditions that occur only in specific situations. There exist many different systems that have been developed to address these issues . One of these systems is the advanced care and alert portable telemedical monitor (AMON), which is capable of measuring an electrocardiogram (ECG), blood oxygen saturation, blood pressure and skin temperature and has integrated software for the real-time processing of the measured health parameters [2]. Another system that was developed is HeartToGo, which can continuously monitor and analyse an ECG in real time in order to detect cardiovascular diseases [3]. And, finally, LifeGuard[5] is a monitoring system, which is capable of measuring ECG, the respiration rate, the blood oxygen saturation, the skin temperature, the heart rate, the blood pressure and body movement.

Currently, Samsung is developing their own eHealth device with the Samsung Simband©. It provides several sensors including an ECG, a Photoplethysmogram (PPG), a Galvanic Skin Response (GSR) sensor, a Bio-Impedance (Bio-Z) sensor, an accelerometer and a thermometer, which are regrouped on a modular platform, which allows to easily integrate more sensors in the future.

A. Challenges

Most of the existing systems [3][5][4] either rely on only one sensor to estimate the state of the patient, or, when using multiple sensors, they still use them individually to detect anomalies in the patient's health. This might lead to erroneous results in the case one or maybe multiple sensors are malfunctioning, which can lead to false positive alarms, which could annoy the patient and make them reluctant to use the device or to false negative alarms, which are dangerous for the well-being of the patient monitored. Some systems tried to overcome this caveat by choosing reliability values for the different sensors [2]. This approach makes use of knowledge on which sensors are more likely to fail than others but fails to take advantage of the data gathered and analyzed by the different sensors. As more and more sensors are developed for wearable devices, the system should be able to use the power of these sensors combined in order to ensure a better monitoring of the patient. Thus, it is important for the system to be capable of mediating between the different sensors.

B. Contribution

Our contribution is twofold: First, we propose a method based on Hidden Markov Models (HMMS) to estimate the risk for different diseases that a patient could have. This is done by analysing the data from each sensor with specially trained HMMS. Secondly, we propose a model for an autonomic system for wearable devices that uses the state estimations from the HMMS in order to mediate between the different conflicting results.

The rest of this paper is structured as follows: The next section shows recent related work that has been done with regards to health care systems. In Section III, we describe the concepts of HMMS and how we use them in order to get different estimations for the state of the patient. In Section IV, we test the accuracy of the sensor analysis using HMMS and present the results. In Section V, we propose a structure for an autonomic system to mediate between the different sensors. Finally, in Section VI, we conclude and discuss future work.

II. Related work

In this section, we will list and describe recent responsive healthcare systems. A very recent healthcare system is shown in [6]. The system has a total of 8 different sensors. However, the main focus of the study was to improve the energy consumption of the whole system and not the classification based on data from different sensors. In [7] and [8] an accelerometer sensor is used in order to help patients with their rehabilitation after a stroke.

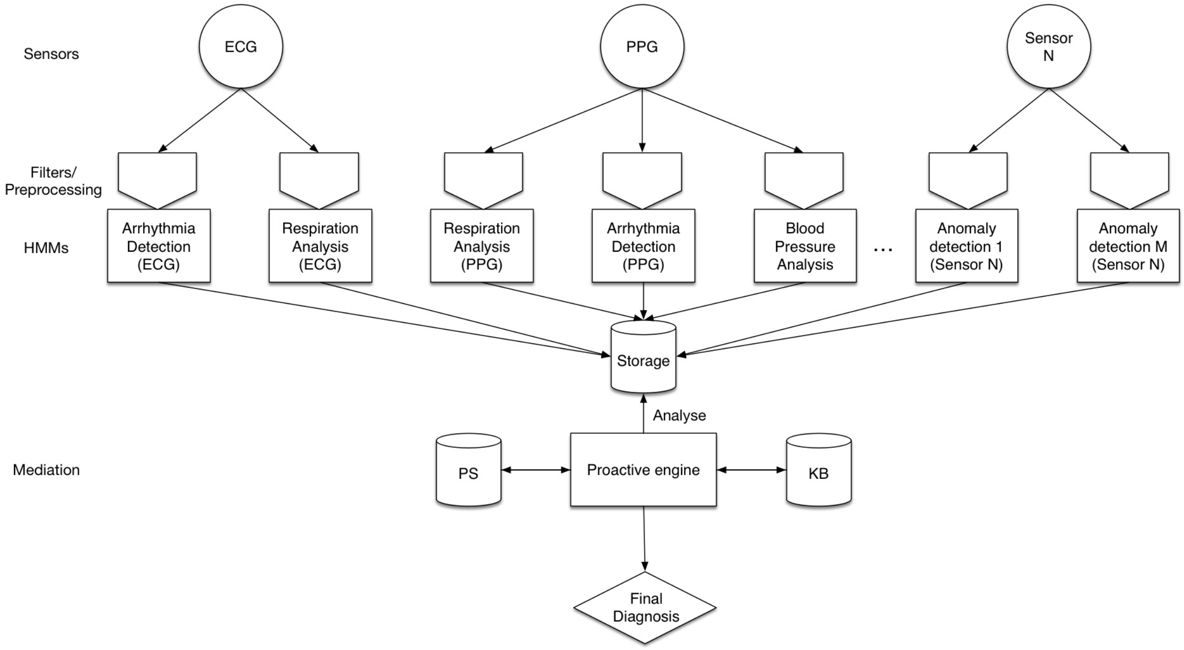

Figure 1: Detailed view of the system

These systems all have in common that they do not use the data from the different sensors together in order to improve the diagnosis. In fact, a recent survey and analysis of existing healthcare systems and applications, by Tsakalakis [9], showed that the current systems are missing the appropriate level of decision support and clinical evaluation. For example, in [10], the authors concentrate on ECG data in order to detect cardiovascular diseases. Another similar approach in order to detect pulse loss based on blood pressure data is presented in [11]. A little bit more sophisticated approach is described in [12] in which the authors not only use an ECG sensor but also an Electroencephalogram (EEG) in order to measure brain activity and an Electrogastrogram (EGG), which records the electrical signals of the muscles in the stomach. However, while they use multiple sensors, the diagnosis is done based on data from individual sensors.

In order to overcome some of these limitations, the authors in [13] proposed a multi-tier hierarchy that uses data from multiple sensors in combination with machine learning methods for disease recognition. In our system, we want to use a rule-based system in addition to machine learning methods to improve the accuracy of the diagnosis.

III. Hidden Markov Models for State Analysis

HMMS have been successfully used in many fields, be it for speech recognition [14][15][16], failure detection [17] or complex action recognitions [18].

Different studies also used them for ECG [19][20][21] and respiration analysis [22]. In this section, we will first give a theoretical overview about HMMS and then we describe how we integrated them into our system. Finally, preliminary test results are presented.

A. *Theoretical background*

An HMM models stochastic sequences as Markov chains where the states are hidden. HMMS consist of five parts [16] :

1) The number of states N in the model. Even though the states are hidden, they generally have a physical meaning. In the case of a patient, they can mean that the patient is in a low, medium, high or no risk state. We denote the individual states as S= $\{S_1, S_2, ..., S_N\}$ and the state at time t as q_t.
2) The number of distinct observation symbols. We denote the individual symbols as V= $\{v_1, v_2, ..., v_M\}$.
3) The state transition probability distribution A = $\{a_ij(k)\}$ where
$\{a_ij(k)\}$ = $P[q_{t+1} = S_j | q_t = S_j], \quad 1 \leq i, j, \leq N.$
4) The observation probability distribution B = $\{b_j(k)\}$ for every state j where
$\{b_j(k)\} = P[v_k \text{ at } t | q_t = S_j], \quad 1 \leq j \leq N, 1 \leq k \leq M.$
5) The initial state distribution $\pi = \{\pi_i\}$ where
$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N.$

There are three fundamental problems for HMMS:

1) Given an observation sequence O= $O_1, O_2 ... O_T$ and a model $\lambda = (A, B, \pi)$, how do we compute $P(O|\lambda)$?
2) Given an observation sequence O= $O_1, O_2 ... O_T$ and a model $\lambda = (A, B, \pi)$, how do we find the most likely state sequence $Q = q_1, q_2, ..., q_T$)?
3) How do we optimise the model parameters A, B and π of the model λ in order to maximize $P(O|\lambda)$?

The solutions to the first and second problems can be both used for classification. With the solution to the first problem, we can calculate the probability that an observation belongs to a specific model. By doing this for different models, we can choose the model with the highest probability as classification.

The second problem can be solved easily by trying every possible state sequence and taking the one with the highest probability. As this method increases exponentially with the length of the observation sequence a more effective solution was developed: the Viterbi decoding algorithm [23] [24]. The Viterbi algorithm calculates the state sequence that has the highest probability to have generated a given observation sequence by only doing subsequent calculations for the partial path with the best probability, thus the complexity only increases linearly with the observation sequence length.

The third problem consists of training the model in such a way that, given a training observation sequence O, the parameters of the model $\lambda = (A, B, \pi)$ are adapted in order to maximise the probability of O given lambda. There does not exist an optimal solution for this problem, but there are several solutions to find local maxima for $P(O|\lambda)$ including the expectation maximisation algorithm [25], the segmental K-means algorithm [26] and the Baum-Welch algorithm [27].

In our system we use the Baum-Welch, also called forward-backward, algorithm in order to train the different HMMS. For the classification phase, we use the Viterbi decoding algorithm.

B. Structure

Some existing approaches already use HMMS in order to estimate the state of equipment [28] by analysing the data of several sensors of the same type simultaneously. In our approach, different HMMS are specialised to detect a specific disease or condition based on the data of different types of sensors, as shown in Figure 1. This means that there are different HMMS that are responsible for detecting the same disease, which in case of sensor malfunctioning can lead to conflicting results, as for example the ECG sensor could detect an arrhythmia while the PPG does not. Independent of possible conflicts, the results of the HMMS will be stored in a database where they wait for further analysis by the rule-based proactive engine, which will be discussed in Section V.

IV. Preliminary Experiments and results

A. Experimental setup

For preliminary tests, we wanted to see how well HMMS can detect arrhythmias. To to this, we used the topology for a HMM described [19] (Figure 2) to distinguish between 3 classes of beats: supra ventricular, normal and ventricular. The individual states correspond to the different stages of a heartbeat. For the training and the testing phase, we used data from the MGH/MF Waveform Database [29][30], which contain three ECG leads and was sampled at a rate of 360 measurements per second as well as an annotation file by expert cardiologist that identified every heartbeat.

1) Initial parameters: Having good initial parameters for HMM is important to get satisfactory classification results [31]. In fact, the starting parameters have to be within one standard deviation [32] from the actual parameters in order to get accurate classification results. This is why, the starting parameters were calculated "manually" with a few samples of each class of heartbeat.

2) Training: Training is done for each class of heartbeat in a specific HMM for the class before they are regrouped in the final HMM described in Figure 2. Training is done using the Baum-Welch algorithm with records from the mgh001 file of the selected dataset.

3) State estimation: For the classification, we use the model from Figure 2. The sequence of states is calculated using the Viterbi algorithm.

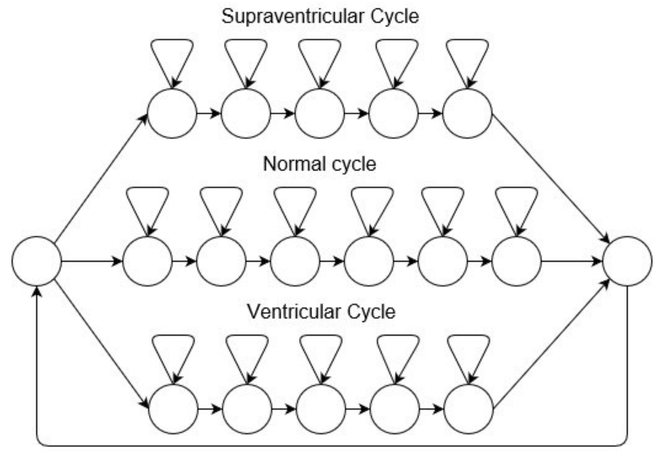

Figure 2: Markov model topology

TABLE I: CLASSIFICATION RESULTS

File	TP	FP	FN	TN	FP rate	Recall	Specificity
mgh001	11	824	1	1747	32.05%	91.67 %	67.95 %
mgh002	333	817	98	4320	15.90 %	77.26 %	84.10%
mgh003	7	1532	0	5449	21.95 %	100 %	78.05%

B. Results

The results are presented in Table I in terms of correctly determined arrhythmias (TP), correctly determined normal heartbeats (TN), false alarms (FP) and not detected arrhythmias (FN).

We see that the recall is actually quite high, as most of the arrhythmias were detected. However, the false positive rate is also high. By analysing the classification results, we found out that the HMM has difficulties to classify the period between heartbeats correctly, meaning that it often considers the period between heartbeats to be an arrhythmia.

The study done in [19] also has quite a lot of false positives depending on the data set used. While the number of false positives are not quite as high, they are still too high to use on an automated wearable system, as the patient would get annoyed really fast and switch the device of if he received notifications of detected problems all the time. In the next section, we propose a model of an autonomic system to try to overcome these difficulties.

V. A Model for an Autonomic eHealth System

A. Autonomic computing

Autonomic computing was introduced in 2001 by IBM [33], in an effort to reduce the need for human involvement in complex computing systems. Shortly afte,r a clearer definition of an autonomic system was developed [34][35]. Autonomic computing is the idea for a system to manage itself and to minimise human intervention. The goals and objectives of the system are ensured by a processing cycle, the MAPE loop, which stands for monitoring, analysing, planning and execution. Also, in order to be classified as an autonomic system, a system needs to exhibit at least the following self-properties, also called self-* properties: self-configuration, self-healing, self-optimisation and self-protection.

B. System structure

The self-managing aspect of these systems is ideal to use for the monitoring of patients on wearable systems. In Figure 4, we see a general overview of the system. As computing power on the wearable devices is quite limited, we consider the wearable device together with a smartphone as one autonomic system. The wearable device collects data with its sensors and forwards them to the smartphone on which the analysis part is done. A more detailed view of the analysis process is shown in Figure 1. It consist of two main steps: in the first step, filters pre-process the data coming from the sensors and pass it to the different HMMS for state analysis. The results from the HMMS are stored in a database, where they are analysed in a second step by a rule-based proactive engine in order to make a final diagnosis.

In the next section, we will discuss the structure of this rule-based engine that implements the properties of an autonomic system. Afterwards, we will then discuss what the job of the different self-properties is in a healthcare setting.

C. Proactive engine

A rule-based proactive engine was developed recently for different platforms (Windows, Android and iOS). Conceptually, the rules run in the engine [36] can be regrouped into scenarios [37] with each scenario regrouping rules that achieve a common goal.

Rules consist of 5 different parts: data acquisition, activation guards, conditions, actions and rule generation and are executed periodically. Both, activation guards and conditions, have to be satisfied in order for a rule to execute its actions. The activation guards are the triggers for a rule to consider taking actions while the conditions are the permissions of a rule. In order to decide, which rules can execute, all rules whose activation conditions are met are first put into a list. In Figure 3, the next steps of the rules' execution process is shown. The rules are split into two categories: diagnosis rules and conflict handling rules. In the first step, the diagnosis rules analyse the state probabilities provided by the HMMS and register appropriate actions that should be taken. As in this case, conflicting data is coming from the ECG and PPG sensor, conflicting actions are registered.

In the second step, the conflict handling rules detect conflicts based on the registered actions and resolve them by giving permissions to the different rules to execute their actions. This is done by calculations based on a chosen priority parameter and the probabilities coming from the HMM. As rules might execute more than one action, permissions are granted for individual actions.

Finally, in the last step the diagnosis rules check their permissions and execute the actions they are allowed to.

D. Self-healing

An autonomic system needs to be able to detect, diagnose and recover from problems occurring inside or also possibly outside the system in order to guarantee an acceptable uptime of the services provided.

In the case of a pervasive healthcare system, the data stream of the sensors might not be complete. For example, due to connection problems, data can be incomplete at times. If the data stream is disconnected for too long, the sensor will be marked as failed.

E. Self-configuration

An autonomic system needs to be able to configure and reconfigure itself in order to adapt itself to different situations, meaning that changes in the internal or external context should not prevent the system of achieving its objective(s).

Self-healing and self-configuration go hand in hand in this system as, as soon as the self-healing module detects problems in the system (in this case, most likely the sensors) and cannot repair them, the self-configuration module needs to adapt the internal parameters of the system in order to take the failures of some sensors into account for the decision making process. Some of the cases in which these two self-properties are used:

1) Failed sensors.
   Completely failed sensors can be detected easily as the stream of data to the system stops.
2) Malfunctioning sensors.
   Malfunctioning sensors can be difficult to detect as their malfunctioning could possibly be interpreted as health problems of the patient.
3) Removal of sensors.
   This should fall under the same category as failed sensors as removed sensors will simply stop sending data to the system.
4) Addition of sensors
   Adding new sensors is a challenge as not only does the system need to detect that there is a new sensor but it also needs to get an accurate description on how to use the data from the new sensor and how it should behave in relation to the data from the other sensors.
5) Recovery of failed or malfunctioning sensors
   The recovery of failed sensors should fall under the same category as adding new sensors.

F. Self-optimisation

An autonomic system always needs to try to improve itself based on different criteria as, for example, execution speed, accuracy, etc.

The self-optimisation module of this system tries to improve the accuracy of the decision making of the system regarding the diagnosis of the patients' illnesses.

This can be done by keeping track of the different decisions made in a Knowledge-Base and by analysing the behaviour of the patient that follows these decisions. Data kept in the Knowledge-Base include the decisions made, as well as the data from the sensors, and/or previous decisions that lead to this decision. For example, after a heart attack diagnosis, the system could check if the patient is still exercising and if his health parameters are stabilised again and conclude that next time it should not make the same diagnosis.

G. Self-protection

An autonomic system needs to be able to anticipate, detect, identify and protect itself from internal and external threats, in order to maintain its integrity and achieve security, privacy and data protection.

In an autonomic healthcare system the self-healing module needs to deal with the following issues:

1) Conflicts with self-healing
   While self-healing tries to keep sensors functioning as

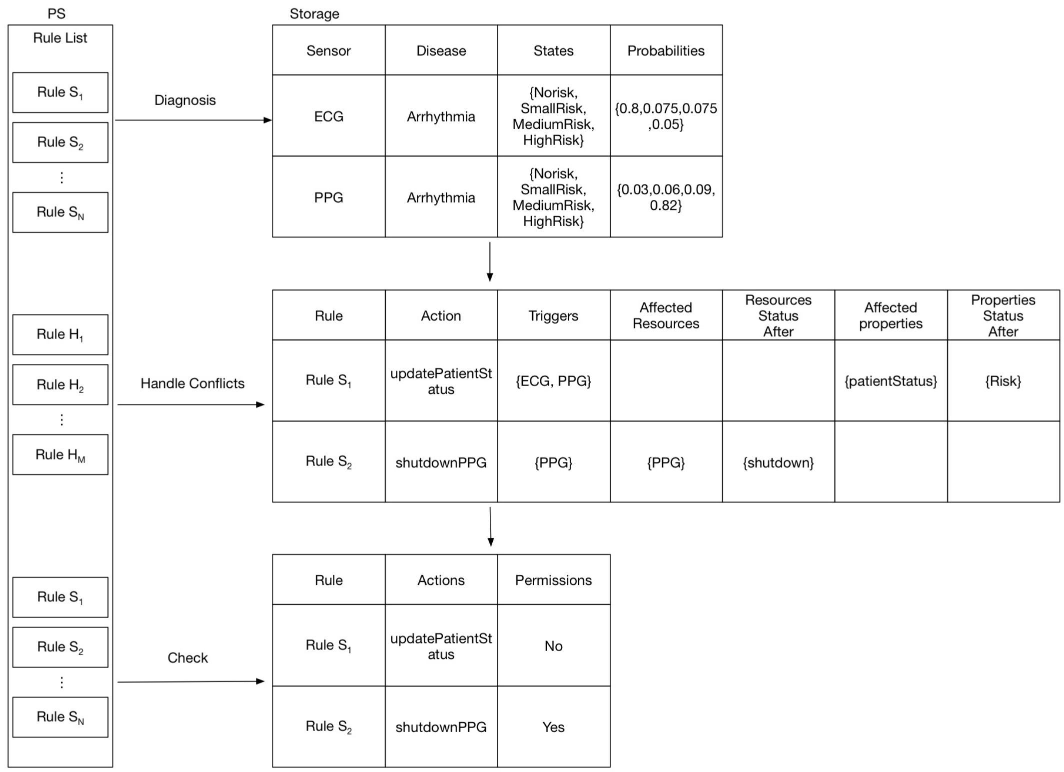

Figure 3: Proactive conflict handling

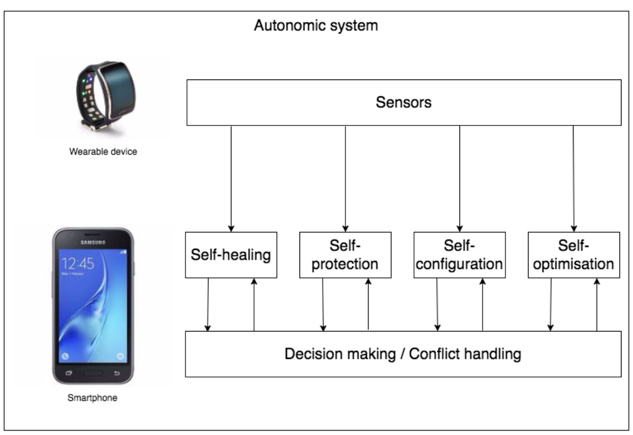

Figure 4: Autonomic healthcare system

long as possible, self-protection rather wants to shut the sensors down than to allow bad data to influence the integrity of the system.

2) Communication between smartphone and wearable device.
Data privacy and integrity is particularly important in E-Health systems. The communication between the wearable and the smartphone thus has to be secure in order to avoid privacy loss and even more importantly, manipulation of a patient's health parameters.

VI. Conclusion and Future work

We proposed an approach using HMMS combined with a rule-based system in order to diagnose the health state of a patient for wearable device. Our tests have shown that the false positive rate of individual HMM is quite high. This is one of the issues we plan to address with our system in the future by mediating between the data coming from different sensors. Additional tests are thus needed to see if the system proposed will be able to rescue the rate of false positives, while maintaining or even also improve the recall. In a second step, we will then also test the accuracy of the classification when one or more sensors are malfunctioning.

Another future work, in order to rule out possible errors related to the initial parameters of the HMM, is to improve the method in how these initial parameters are obtained. While we may, to some extent, rely on experts to provide this initial parameters estimation it would be more reliable to have the training algorithm adapt itself. In [38], Won et al. use genetic algorithms, in conjunction with the standard training algorithm for HMMS, in order to explore different starting conditions. Their study has shown that the addition of genetic algorithms, even with a naive implementation, lead to slightly superior classification results. We think that even slight improvements are important in a healthcare setting and plan to improve the training of the HMMS with a genetic algorithm.

References

[1] G. Kortuem and Z. Segall, "Wearable communities: augmenting social networks with wearable computers," IEEE Pervasive Computing, vol. 2, no. 1, 2003, pp. 71–78.

[2] U. Anliker et al., "Amon: a wearable multiparameter medical monitoring and alert system," IEEE transactions on information technology in biomedicine, vol. 8, no. 4, 2004, pp. 415–427.

[3] Z. Jin, J. Oresko, S. Huang, and A. C. Cheng, "Hearttogo: a personalized medicine technology for cardiovascular disease prevention and detection," in Life Science Systems and Applications Workshop, 2009. LiSSA 2009. IEEE/NIH. IEEE, 2009, pp. 80–83.

[4] R. A. Dobrican and D. Zampunieris, "A proactive solution, using wearable and mobile applications, for closing the gap between the rehabilitation team and cardiac patients," in Healthcare Informatics (ICHI), 2016 IEEE International Conference on. IEEE, 2016, pp. 146–155.

[5] C. W. Mundt et al., "A multiparameter wearable physiologic monitoring system for space and terrestrial applications," IEEE Transactions on Information Technology in Biomedicine, vol. 9, no. 3, 2005, pp. 382–391.

[6] A. M. Nia, M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Energy-efficient long-term continuous personal health monitoring," IEEE Transactions on Multi-Scale Computing Systems, vol. 1, no. 2, 2015, pp. 85–98.

[7] S. Patel et al., "A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology," Proceedings of the IEEE, vol. 98, no. 3, 2010, pp. 450–461.

[8] T. Hester et al., "Using wearable sensors to measure motor abilities following stroke," in Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on. IEEE, 2006, pp. 4–pp.

[9] M. Tsakalakis and N. G. Bourbakis, "Health care sensorâĂŤbased systems for point of care monitoring and diagnostic applications: A brief survey," in Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE. IEEE, 2014, pp. 6266–6269.

[10] J. J. Oresko et al., "A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing," IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 3, 2010, pp. 734–740.

[11] J. Rickard, S. Ahmed, M. Baruch, B. Klocman, D. O. Martin, and V. Menon, "Utility of a novel watch-based pulse detection system to detect pulselessness in human subjects," Heart Rhythm, vol. 8, no. 12, 2011, pp. 1895–1899.

[12] F.-S. Jaw, Y.-L. Tseng, and J.-K. Jang, "Modular design of a long-term portable recorder for physiological signals," Measurement, vol. 43, no. 10, 2010, pp. 1363–1368.

[13] Y. Hongxu and N. Jha, "A hierarchical health decision support system for disease diagnosis based on wearable medical sensors and machine learning ensembles," IEEE Transactions on Multi-Scale Computing Systems, 2017.

[14] X. D. Huang, Y. Ariki, and M. A. Jack, Hidden Markov models for speech recognition. Edinburgh university press Edinburgh, 1990, vol. 2004.

[15] M. Gales and S. Young, "The application of hidden markov models in speech recognition," Foundations and trends in signal processing, vol. 1, no. 3, 2008, pp. 195–304.

[16] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, 1989, pp. 257–286.

[17] F. Salfner and M. Malek, "Using hidden semi-markov models for effective online failure prediction," in Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on. IEEE, 2007, pp. 161–174.

[18] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in Computer vision and pattern recognition, 1997. proceedings., 1997 ieee computer society conference on. IEEE, 1997, pp. 994–999.

[19] D. A. Coast, R. M. Stern, G. G. Cano, and S. A. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," IEEE Transactions on biomedical Engineering, vol. 37, no. 9, 1990, pp. 826–836.

[20] S. Hu, Z. Shao, and J. Tan, "A real-time cardiac arrhythmia classification system with wearable electrocardiogram," in Body Sensor Networks (BSN), 2011 International Conference on. IEEE, 2011, pp. 119–124.

[21] R. V. Andreão, B. Dorizzi, and J. Boudy, "Ecg signal analysis through hidden markov models," IEEE Transactions on Biomedical engineering, vol. 53, no. 8, 2006, pp. 1541–1549.

[22] T. Al-Ani, Y. Hamam, R. Fodil, F. Lofaso, and D. Isabey, "Using hidden markov models for sleep disordered breathing identification," Simulation Modelling Practice and Theory, vol. 12, no. 2, 2004, pp. 117–128.

[23] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE transactions on Information Theory, vol. 13, no. 2, 1967, pp. 260–269.

[24] G. D. Forney, "The viterbi algorithm," Proceedings of the IEEE, vol. 61, no. 3, 1973, pp. 268–278.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," Journal of the royal statistical society. Series B (methodological), 1977, pp. 1–38.

[26] B.-H. Juang and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden markov models," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 9, 1990, pp. 1639–1641.

[27] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology," Bull. Amer. Math. Soc, vol. 73, no. 3, 1967, pp. 360–363.

[28] M. Dong and D. He, "Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," European Journal of Operational Research, vol. 178, no. 3, 2007, pp. 858–878.

[29] J. Welch, P. Ford, R. Teplick, and R. Rubsamen, "The massachusetts general hospital-marquette foundation hemodynamic and electrocardiographic database–comprehensive collection of critical care waveforms," Clinical Monitoring, vol. 7, no. 1, 1991, pp. 96–97.

[30] A. L. Goldberger et al., "Physiobank, physiotoolkit, and physionet," Circulation, vol. 101, no. 23, 2000, pp. e215–e220.

[31] L. R. Rabiner, B.-H. Juang, S. Levinson, and M. Sondhi, "Some properties of continuous hidden markov model representations," AT&T technical journal, vol. 64, no. 6, 1985, pp. 1251–1270.

[32] D. A. Coast, Cardiac arrhythmia analysis using hidden Markov models. UMI, 1988.

[33] A. C. Manifesto, "Ibms perspective on the state of information technology," http://www.research.ibm.com/autonomic/manifesto/, 2001.

[34] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," Computer, vol. 36, no. 1, 2003, pp. 41–50.

[35] P. Lalanda, J. A. McCann, and A. Diaconescu, Autonomic computing. Springer, 2013.

[36] D. Zampunieris, "Implementation of efficient proactive computing using lazy evaluation in a learning management system (extended version)," International Journal of Web-Based Learning and Teaching Technologies, vol. 3, 2008, pp. 103–109.

[37] D. Shirnin, S. Reis, and D. Zampunieris, "Design of proactive scenarios and rules for enhanced e-learning," in Proceedings of the 4th International Conference on Computer Supported Education, Porto, Portugal 16-18 April, 2012. SciTePress–Science and Technology Publications, 2012, pp. 253–258.

[38] K.-J. Won, A. Prügel-Bennett, and A. Krogh, "Training hmm structure with genetic algorithm for biological sequence analysis," Bioinformatics, vol. 20, no. 18, 2004, pp. 3613–3619.