ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΤΩΝ

Software Design and Optimization of ECG Signal Analysis and Diagnosis for Embedded IoT Devices

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αζαριάδη Δήμητρα

Επιβλέπων : Δημήτριος Ι. Σούντρης
Αναπληρωτής Καθηγητής

Αθήνα, Φεβρουάριος 2016

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΤΩΝ

Software Design and Optimization of ECG Signal Analysis and Diagnosis for Embedded IoT Devices

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αζαριάδη Δήμητρα

Επιβλέπων : Δημήτριος Ι. Σούντρης
Αναπληρωτής Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Φεβρουαρίου 2016.

.....	.....	.....
Δημήτριος Ι. Σούντρης	Κιαμάλ Πεκμεστζή	Κωνσταντίνα Νικήτα
Αναπληρωτής Καθηγητής	Καθηγητής	Καθηγήτρια

Αθήνα, Φεβρουάριος 2016

.....
Αζαριάδη Δήμητρα

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Contents

Σύντομη Περίληψη

Στο πρώτο μέρος αυτής της εργασίας, σχεδιάζουμε τον αλγόριθμο για την ανάλυση και ταξινόνηση του σήματος ΗΚΓ, σε περιβάλλον MATLAB. Η βάση δεδομένων που έχει επιλεγεί ως πηγή για ΗΚΓ είναι η MIT-BIH Arrhythmia Database. Η δομή του αλγορίθμου αποτελείται από τα ακόλουθα στάδια: φιλτράρισμα, ανίχνευση παλμών, κατακερματισμό σε παλμούς, εξαγωγή χαρακτηριστικών, ταξινόμηση. Η είσοδος είναι ένα σήμα ΗΚΓ, φιλτράρεται, εντοπίζονται οι καρδιακοί παλμοί που περιλαμβάνονται σε αυτό, το σήμα χωρίζεται σε μεμονωμένους παλμούς, εξάγονται τα χαρακτηριστικά γνωρίσματα για κάθε παλμό, και η έξοδος του τελικού σταδίου είναι η κατάταξη για κάθε παλμό ως 'Normal' ή 'Abnormal'. Για την υλοποίηση αυτών των σταδίων, χρησιμοποιούνται MATLAB built-in συναρτήσεις, η βιβλιοθήκη LIBSVM για τον ταξινομητή SVM που χρησιμοποιείται, καθώς επίσης και συναρτήσεις που παρέχονται από τη βάση δεδομένων στο WFDB Matlab toolkit. Το στάδιο της ταξινόμησης αποτελείται από έναν ταξινομητή SVM, μια μέθοδο εποπτευόμενης μηχανικής μάθησης. Χρησιμοποιούμε αρχεία σχολιαμού που περιλαμβάνονται στη βάση δεδομένων, τα οποία περιλαμβάνουν διάγνωση για κάθε παλμό που περιέχεται σε ένα αρχείο ΗΚΓ, που έχει πραγματοποιηθεί από γιατρούς. Το πρόβλημα που εντοπίστηκε σε αυτό το σημείο της ανάλυσης, είναι ότι υπάρχει μια αναντιστοιχία των καρδιακών παλμών που ανιχνεύονται σε μια ηχογράφηση από τις συναρτήσεις που παρέχονται στην εργαλειοθήκη, με τους διαγνωσμένους παλμούς από τους γιατρούς. Αυτό συμβαίνει επειδή οι ανιχνευτές παλμών αδυνατούν να εντοπίσουν όλους τους παλμούς και μερικές λανθασμένες ανιχνεύσεις λαμβάνουν επίσης μέρος. Το πρόβλημα αυτό ξεπεράστηκε με τη διαμόρφωση μιας διαδικασίας που μας επιτρέπει να αντιστοιχίσουμε τους σωστά ανιχνευμένους παλμούς με τις αντίστοιχες διαγνώσεις στα αρχεία σχολιασμών. Στη συνέχεια, μπορούμε να εκτελέσουμε μια εξερεύνηση στο χώρο σχεδιασμού για τα διαφορετικά χαρακτηριστικά γνωρίσματα που εξάγονται από το σήμα στο στάδιο της εξαγωγής χαρακτηριστικών. Χρησιμοποιούμε Discrete Wavelet Transform ως μέθοδο εξαγωγής χαρακτηριστικών. Τα χαρακτηριστικά αυτά χρησιμεύουν ως είσοδος για το στάδιο της ταξινόμησης. Οι μετρικές που χρησιμοποιούνται για να επιλέξει ο σχεδιαστής το καλύτερο design είναι η ακρίβεια και η υπολογιστικό κόστος του σταδίου ταξινόμησης.

Στο δεύτερο μέρος της ανάλυσης, προτείνουμε την προσθήκη ενός επιπλέον σταδίου στην αλγοριθμική δομή. Αυτό το στάδιο είναι τοποθετημένο ακριβώς πριν το τελικό στάδιο της ταξινόμησης, και αποτελείται από ένα ταξινομητή SVM που θα λαμβάνει ως είσοδο τα χαρακτηριστικά γνωρίσματα που εξάγονται κατά το προηγούμενο στάδιο και θα ταξινομεί τους εντοπισμένους παλμούς ως αληθείς ή ψευδείς ανιχνεύσεις. Κάθε αληθινή ανίχνευση θα συνεχίσει μέχρι το τελικό στάδιο, ενώ κάθε ψευδής ανίχνευση θα πρέπει να απορρίπτεται. Μια εξερεύνηση στο χώρο σχεδιασμού γίνεται ομοίως όπως έγινε στην αρχική δομή.

Στο τελευταίο μέρος της ανάλυσης, η αρχική αλγοριθμική ροή υλοποιείται στην ενσωματωμένη πλατφόρμα Galileo της Intel. Για να γίνει αυτό, ο αλγόριθμος μετατρέπεται σε C κώδικα. Στην τελική του μορφή, το πρόγραμμα διαβάζει δείγμα-δείγμα ένα ψηφιοποιημένη στα 360 δείγματα ανά δευτερόλεπτο σήμα ΗΚΓ, και η ροή ανάλυση εκτελείται για κάθε σύνολο 3000 δειγμάτων που διαβάζεται. Οι 10 καλύτερες σχεδιστικές επιλογές, οι 10 πιο απαιτητικές σε υπολογιστικό κόστος, καθώς και 11 ενδιάμεσες επιλογές, όπως προέκυαν από την εξερεύνηση του χώρου σχεδιασμού στο πρώτο μέρος της εργασίας, υλοποιήθηκαν στο Galileo. Οι ακρίβειες που ε-

πιτεύχθηκαν ήταν πάνω από ικανοποιητικές και το υπολογιστικό κόστος ήταν τέτοια, ώστε η ανάλυση ΗΚΓ και ταξινόμηση να μπορεί να εκτελεσθεί σε πραγματικό χρόνο.

Λέξεις κλειδιά: Ανάλυση ECG, Ταξινόμηση καρδιακού παλμού, Support-Vector-Machine (SVM), Discrete-Wavelet-Transform (DWT), Internet-of-Things (IoT), Ενσωματωμένα συστήματα

Abstract

In the first part of this work, we design the algorithm for ECG signal analysis and classification and implement it in Matab environment. The database chosen as source for ECG recordings is the MIT-BIH Arrhythmia database, provided by PhysioNet. The structure of the algorithm consists of the following stages: filtering, heartbeat detection, heartbeat segmentation, feature extraction, classification. The input is an ECG signal, it is filtered, the heartbeats included in it are detected, the signal is segmented into beats, features are extracted from each beat, and the output of the final stage is a label (Normal or Abnormal) for each heartbeat. For the implementation of these stages we used Matlab built-in functions, the LIBSVM library for the SVM classifier used, and also functions provided by PhysioNet in the WFDB Matlab toolkit. The classification stage consists of a SVM classifier, a supervised machine learning method. We use annotations files included in the database, which provide diagnosis labeling of each heartbeat included in each ECG recording, done by doctors. The problem detected at this point of the analysis, is that there is a mismatch in the heartbeats detected in a recording by the functions provided in the toolkit, and the heartbeats annotated by the doctors. This happens because heartbeat detectors fail to detect all heartbeats and some false detections also take place. We overcome this problem by forming a procedure that allows us to match the correctly detected heartbeats with their corresponding labels in the annotation files. Next, we perform a design space exploration over different features extracted from the signal in the feature extraction stage. We use discrete wavelet transform as feature extraction method. These features serve as input for the classification stage. The metrics used to decide upon the best design are the accuracy and computational cost of the classification stage.

In the second part of the analysis, we suggest the addition of an extra stage to the algorithmic structure. This stage is placed right before the final classification stage, and consists of an SVM classifier that would take as input the features extracted in the previous stage and classify the heartbeats as true or false detections. True beats will continue to the final stage, while false beats will be discarded. A design space exploration is performed similarly as done in the initial structure.

In the last part of the analysis, the initial algorithmic flow is implemented on the Intel IoT based Galileo board. To do so, the algorithm is converted in C code. In its final form, the program reads sample by sample a digitized at 360 samples per second ECG signal, and the analysis flow is executed for every set of 3000 samples that is read. The 10 best configurations, the 10 most demanding in computational cost configurations, as well as 11 configurations from inbetween, as resulted from the design space exploration in the first part of the work, were implemented on the Galileo board. The accuracies achieved were above satisfactory, and the computational cost was such so that the ECG analysis and classification can be performed in real-time.

Keywords: ECG analysis, ECG heartbeat classification, Support-Vector-Machine (SVM), Discrete-Wavelet-Transform (DWT), Internet-of-Things (IoT), Embedded Systems

Εκτεταμένη Περίληψη

Οι καρδιαγγειακές παθήσεις, η κύρια αιτία του πρόωρου θανάτου στον κόσμο, είναι μια ομάδα ασθενειών που αφορούν στην καρδιά ή τα αιμοφόρα αγγεία. Περιλαμβάνουν ασθένειες της στεφανιαίας αρτηρίας όπως η στηθάγχη και το έμφραγμα του μυοκαρδίου, κοινώς γνωστό ως καρδιακή προσβολή, την υπερτασική καρδιακή νόσο, ρευματική καρδιοπάθεια, μυοκαρδιοπάθεια, κολπική μαρμαρυγή, ενδοκαρδίτιδα, αορτικά ανευρύσματα, περιφερική αρτηριακή νόσο και φλεβική θρόμβωση. Περισσότεροι άνθρωποι πεθαίνουν κάθε χρόνο από καρδιαγγειακές παθήσεις, παρά από οποιαδήποτε άλλη αιτία, καθιστώντας τη ως τη νούμερο ένα αιτία θανάτου σε παγκόσμιο επίπεδο. Ταυτόχρονα, καθώς ο παγκόσμιος πληθυσμός ξεπερνά τα 7 δισεκατομμύρια και περισσότεροι άνθρωποι φτάνουν στη τρίτη ηλικία, ο αριθμός των θανάτων από καρδιαγγειακές νόσους εξακολουθεί να είναι σε άνοδο. Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας, σε παγκόσμιο επίπεδο, ο αριθμός των θανάτων που οφείλονται σε καρδιαγγειακές παθήσεις αυξήθηκε κατά 41% μεταξύ του 1990 και του 2013, δηλαδή από 12,3 εκατομμύρια θανάτους σε 17,3 εκατομμύρια θανάτους. Στις Ηνωμένες Πολιτείες το 11% των ανθρώπων μεταξύ 20 και 40 έχουν καρδιαγγειακά νοσήματα, ενώ το 37% μεταξύ 40 και 60, το 71% των ατόμων μεταξύ 60 και 80, και το 85% των ανθρώπων άνω των 80 ετών. Αυτά τα υψηλά ποσοστά μαζί με τον αυξανόμενο ρυθμό θνησιμότητας, κάνουν επιτακτική την ανάγκη για στενή και συνεχή ιατρική παρακολούθηση και φροντίδα των ασθενών που πάσχουν από τέτοιες ασθένειες.
Η ανάλυση του ηλεκτροκαρδιογραφήματος (ΗΚΓ) έχει χρησιμοποιηθεί εκτεταμένα για τη διάγνωση πολλών καρδιακών ασθενειών. Η αυτόματη και ακριβής αναγνώριση των αρρυθμιών σε ένα ηλεκτροκαρδιογράφημα είναι απαραίτητη για τη σωστή θεραπεία του ασθενούς. Τα τελευταία χρόνια, διάφοροι αλγορίθμοι έχουν αναπτυχθεί για την ανάλυση και την ταξινόμηση του ΗΚΓ. Η πλειοψηφία αυτών των αλγορίθμων ακολουθούν την εξής δομή: ανίχνευση των καρδιακών παλμών σε ένα ΗΚΓ, εφαρμογή μεθόδων εξαγωγής χαρακτηριστικών γνωρισμάτων στους παλμούς, και χρήση μεθόδων ταξινόμησης για την εξαγωγή της διάγνωσης.
Με την ανάπτυξη της τεχνολογίας, οι φορετές συσκευές (wearable devices) παίζουν σημαντικό ρόλο στην καταγραφή του ΗΚΓ. Έτσι, η έρευνα που διεξάγεται στον τομέα της αυτόματης ανάλυσης ΗΚΓ έχει, σε ένα μεγάλο μέρος, προσανατολιστεί στην ανάπτυξη αποδοτικών αλγορίθμων που ξεπερνούν τους περιορισμούς των ενσωματωμένων φορητών και φορετών συσκευών. Επιπλέον, αυτές οι συσκευές προορίζονται συνήθως ως μέρος ενός δικτύου, όπου το ΗΚΓ ανιχνεύεται από φορητούς αισθητήρες, στη συνέχεια διαβιβάζεται στη φορετή συσκευή όπου επεξεργάζεται και αναλύεται, και τελικά στέλνεται σε μια απομακρυσμένη συσκευή για περαιτέρω ανάλυση και αποθήκευση. Αυτό το δίκτυο αισθητήρων, υπολογιστικών συσκευών και συσκευών μετάδοσης πληροφορίας αναφέρεται ως Internet of Things (IoT). Σε αυτό το πλαίσιο, το αντικείμενο της παρούσας διπλωματικής εργασίας είναι να αναπτύξουμε λογισμικό που θα υποστηρίζει την ανάλυση του ΗΚΓ για την εξαγωγή χαρακτηριστικών και τεχνικές ταξινόμησης για τη διάγνωση της καρδιακής πάθησης. Η υλοποίηση γίνεται πάνω στην ενσωματωμένη πλατφόρμα Galileo της Intel, η οποία είναι μία από τις προτεινόμενες συσκευές IoT της Intel.
Στο πρώτο μέρος αυτής της εργασίας, σχεδιάζουμε τον αλγόριθμο για την ανάλυση και ταξινόμηση του ΗΚΓ, και η υλοποίηση γίνεται σε περιβάλλον Matlab. Η βάση δεδομένων που χρησιμοποιήθηκε ως πηγή για ΗΚΓ είναι η MIT-BIH Arrhythmia database, που παρέχεται

από τη PhysioNet. Η βάση δεδομένων αποτελείται από 48 ημίωρα αποσπάσματα ΗΚΓ από δύο κανάλια. Τα δεδομένα έχουν φιλτραριστεί με ζωνοπερατό φίλτρο στα 0.1-100Hz και η ψηφιοποίηση έχει γίνει σε 360 δείγματα ανά δευτερόλεπτο ανά κανάλι. Από τα 48 ΗΚΓ, στα πλαίσια της παρούσας ανάλυσης, χρησιμοποιήθηκαν οι καταγραφές του πρώτου καναλιού από τα 45 ΗΚΓ, που αντιστοιχούν σε απαγωγή MLII. Η βάση δεδομένων παρέχει επίσης σημειώσεις για κάθε ΗΚΓ, όπου καρδιολόγοι έχουν τοποθετήσει μια ετικέτα διάγνωσης για κάθε παλμό που περιλαμβάνεται στο ΗΚΓ. Ο πίνακας 1 δείχνει τους τύπους παλμών που περιέχονται στη βάση δεδομένων και την αντιστοίχισή τους στους τύπους παλμών όπως προτείνονται από τον οργανισμό American Heart Association (AHA), και ο πίνακας 2 δείχνει τα ποσοστά ετικετών της βάσης που αντιστοιχούν σε κάθε τύπο παλμού.

Table 1: Αντιστοίχιση των τύπων παλμών της MIT-BIH βάσης με της ομάδες παλμών του AHA

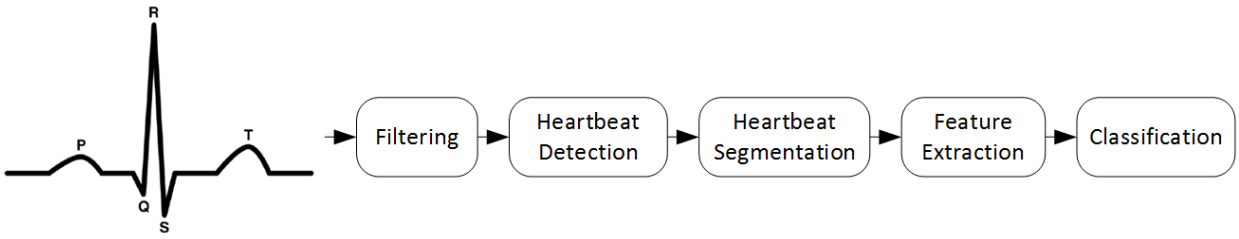
AHA heartbeat class	**N**	**V**	**F**	**E**	**P**	**Q**	**O**
MIT-BIH heartbeat types	**N (normal)**	**V (premature ventricular contraction)**	**F (fusion of ventricular and normal)**	**E (ventricular escape)**	**P (paced)**	**Q (unclassifiable)**	**! (ventricular flutter wave)**
	L (left bundle branch block)		**f (fusion of paced and normal)**				**P (non-conducted P wave)**
	R (right bundle branch block)						
	A (atrial premature)						
	a (aberrated atrial premature)						
	J (nodal premature)						
	S (supraventricular contraction)						
	e (atrial escape)						
	j (nordal escape)						

Το σχήμα 1 απεικονίζει τη δομή του αλγορίθμου ανάλυσης και ταξινόμησης ΗΚΓ. Η πρώτη απαγωγή του ψηφιοποιημένου ΗΚΓ εφαρμόζεται ως είσοδος στο σύστημα. Για την υλοποίηση αυτών των σταδίων χρησιμοποιούνται built-in συναρτήσεις του Matlab, η βιβλιοθήκη LIBSVM

Table 2: Αριθμός παλμών της βάσης που αντιστοιχούν σε κάθε ομάδα

Heartbeat class	N	V	F	E	P	Q	O	total
number of beats	93411	7129	1785	106	7028	33	665	110157
% of total beats	84.8	6.47	1.62	0.096	6.38	0.03	0.6	100

για τον SVM classifier που χρησιμοποιείται, καθώς επίσης και συναρτήσεις που παρέχονται από τη PhysioNet στο WFDB Matlab toolkit.



Σχήμα 1: Προτεινόμενη ροή ανάλυσης

1. Filtering. Μια μονάδα φιλτραρίσματος λειτουργεί ως στάδιο προεπεξεργασίας, προκειμένου να αφαιρεθεί ο θόρυβος από το σήμα. Χρησιμοποιήθηκε η συνάρτηση filter() του Matlab για την εγαρμογή ενός ζωνοπερατού φίλτρου στα 1-50Hz.

2. Heartbeat Detection. Το φιλτραρισμένο σήμα στη συνέχεια διαβιβάζεται στη μονάδα ανίχνευσης παλμών, η οποία προσπαθεί να εντοπίσει όλους τους καρδιακούς παλμούς που περιέχονται στο σήμα εισόδου. Για το σκοπό αυτό χρησιμοποιούνται οι συναρτήσεις wqrs() και ecgpuwave() του WFDB Matlab toolkit. Πρώτα, η συνάρτηση wqrs εφαρμόζεται στο σήμα, η οποία μας δίνει τις θέσεις όλων των συμπλέγματα QRS που βρέθηκαν στο σήμα. Αυτή η πληροφορία μαζί με το σήμα, είναι η είσοδος για την ecgpuwave, η οποία μας δίνει την ακριβή θέση όλων των κορυφών R που βρέθηκαν στο σήμα. Η ανίχνευση του QRS, και ειδικά η ανίχνευση του κύματος R στο ΗΚΓ είναι πιο εύκολη από ότι για άλλα τμήματα του σήματος λόγω της μορφής και του υψηλού του πλάτους. Κάθε ανίχνευση κορυφής R αντιστοιχεί στην ανίχνευση ενός καρδιακού παλμού.

3. Heartbeat Segmentation. Ακολουθεί η μονάδα κατακερματισμού, όπου το σήμα εισόδου χωρίζεται σε μεμονομένους καρδιακούς παλμούς, σύμφωνα με τις πληροφορίες που προέρχονται από το προηγούμενο στάδιο. Αφού εντοπίστηκαν οι κορυφές R για κάθε παλμό, μπορούμε να προχωρήσουμε στο διαμερισμό του ΗΚΓ σε μεμονομένους καρδιακούς παλμούς. Για να γίνει αυτό, θα πρέπει να αποφασίσουμε σχετικά με ένα παράθυρο, το οποίο έχοντας ως κέντρο τη θέση της κορυφής R, θα καλύπτει ολόκληρη τη κυματομορφή του καρδιακού παλμού. Επιλέγουμε ως πλάτος παραθύρου τα 257 δείγματα.

4. Feature Extraction. Προκειμένου να επιτευχθεί μεγαλύτερη απόδοση ταξινόμησης, μια μονάδα εξαγωγής χαρακτηριστικών είναι το επόμενο στάδιο. Εκεί, για κάθε παραγόμενο παλμό, εξάγεται ένα διάνυσμα χαρακτηριστικών, που περιέχει ένα μικρότερο αριθμό στοιχείων από τα δείγματα ΗΚΓ που σχηματίζουν τον καρδιακό παλμό. Ως μηχανισμό εξαγωγής χαρακτηριστικών, χρησιμοποιούμε Discrete Wavelet Transform (DWT). Ως βάση χρησιμοποιείται Daubechies τάξης 2 (db2), και πραγματοποιούνται 4 επίπεδα αποσύνθεσης. Η συνάρτηση wavedec() του Matlab χρησιμοποιήθηκε για την αποσύνθεση του σήματος, και οι συναρτήσεις appcoef() και detcoef() για την εξαγωγή των approximation και detail coefficients, αντίστοιχα. Από τη διαδικασία αυτή παράγωνται 8 σετ από coefficients. Επειδή η κυματομορφή του παλμού αποτελείται από 257 δείγματα, ο αριθμός των coefficients για το πρώτο, δεύτερο, τρίτο και τέταρτο επίπεδο είναι αντίστοιχα 130, 66, 34, 18. Άρα, για κάθε παλμό παίρνουμε 494 coefficients. Τελικά, το διάνυσμα που χρησιμεύει ως είσοδος για το στάδιο της ταξινόμησης προκύπτει από design space exploraation πάνω σε όλους τους συνδυασμούς των 8 αυτών σετ.

5. Το στάδιο της ταξινόμησης αποτελείται από έναν ταξινομητή Support Vector Machine (SVM), μια επιβλεπόμενη μέθοδο μηχανικής μάθησης. Ο ταξινομητής αυτός χαρκτηρίζει κάθε παλμό είτε ως 'Normal' ή 'Abnormal'. Προκειμένου να παραχθεί το μοντέλο του SVM, θα πρέπει να δημιουργήσουμε ένα σύνολο δεδομένων εκπαίδευσης και ένα σύνολο επαλήθευσης, χρησιμοποιώντας τα 45 επιλεγμένα αρχεία ΗΚΓ από τη βάση δεδομένων. Όπως περιγράψαμε προηγουμένως, όλα τα αρχεία ΗΚΓ πρώτα φιλτραριστήκαν, εντοπίστηκαν οι R κορυφές και στη συνέχεια τα σήματα χωρίστηκαν σε μεμονομένους παλμούς, διαμορφώνοντας ένα σύνολο 104.581 καρδιακών παλμών. Για κάθε παλμό χρειαζόμαστε ένα διάνυσμα χαρακτηριστικών και μια τιμή-στόχο. Το διάνυσμα χαρακτηριστικών αποτελείται από τα DWT coefficients που εξήχθησαν στο προηγούμενο στάδιο. Ως τιμή-στόχο, χρησιμοποιούμε τις ετικέτα που παρέχονται από τους γιατρούς για κάθε παλμό.

Το πρόβλημα που εντοπίστηκε σε αυτό το σημείο της ανάλυσης, είναι ότι υπάρχει μια αναντιστοιχία των καρδιακών παλμών που ανιχνεύονται σε ένα ΗΚΓ από τις συναρτήσεις που παρέχονται στο WFDB Matlab toolkit, και αυτών που υπάρχουν σχολιασμένοι από τους γιατρούς. Αυτό συμβαίνει επειδή οι ανιχνευτές καρδιακού παλμού αδυνατούν να ανιχνεύσουν όλους τους παλμούς και επίσης πραγματοποιούνται και μερικές λανθασμένες ανιχνεύσεις. Στην εικόνα 2, οι μάυροι κύκλοι δείχνουν τους παλμούς όπως έχουν καταγραφεί από τους γιατρους, ενώ οι υπόλοιποι όπως εντοπίστηκαν από τις συναρτήσεις, και κατατάσονται σε: True, False και Missed. Προκειμένου να αποφασίσουμε σε ποια από τις τρεις αυτές κατηγορίες ανήκει κάθε εντοπισμένος παλμός, ορίζουμε μία απαιτούμενη απόσταση από τους πραγματικούς παλμούς που έχουν εντοπιστεί από τους γιατρούς. Για το σκοπό αυτό διαξάγουμε μια στατιστική μελέτη πάνω σε όλα τα δεδομένα της βάσης, για την εκτίμηση της χρονικής διάρκειας μεταξύ δύο διαδοχικών παλμών και της διάρκειας του συμπλέγματος QRS. Τα αποτελέμσατα σε seconds φαίνονται στην εικόνα 3. Είναι προφανές ότι η τιμή που παρουσιάζει την πιο ισχυρή συμπεριφορά είναι το σύμπλεγμα QRS και ως εκ τούτου η μέση τιμή του, η οποία είναι περίπου 0,11 δευτερόλεπτα θα χρησιμοποιηθεί για τον προσδιορισμό του ζητούμενου κατωφλίου T. Για δειγματοληψία σε συχνότητα fs = 360Hz, το σύμπλεγμα QRS καταλαμβάνει περίπου 40 δείγματα δεδομένων. Ορίζουμε το κατώφλι T ως το μισό της διάρκεια του συμπλέγματος QRS, δηλαδή 20 δείγματα. Στη συνέχεια, η παρακάτω διαδικασία χρησιμοποιήθηκε για το χαρακτηρισμό των εντοπισμένων παλμών, και την αντιστοίχησή τους με κάποια ετικέτα. Με τη διαδικασία αυτή αντιστοιχίζουμε τους True παλμούς με τις ετικέτες τους στα αρχεία σχολιασμού. Στους False παλμούς αποδίδουμε ετικέτα 'Abnoral'. Για την επιβαιβέωση της επιλογής αυτής, εκπαιδεύουμε

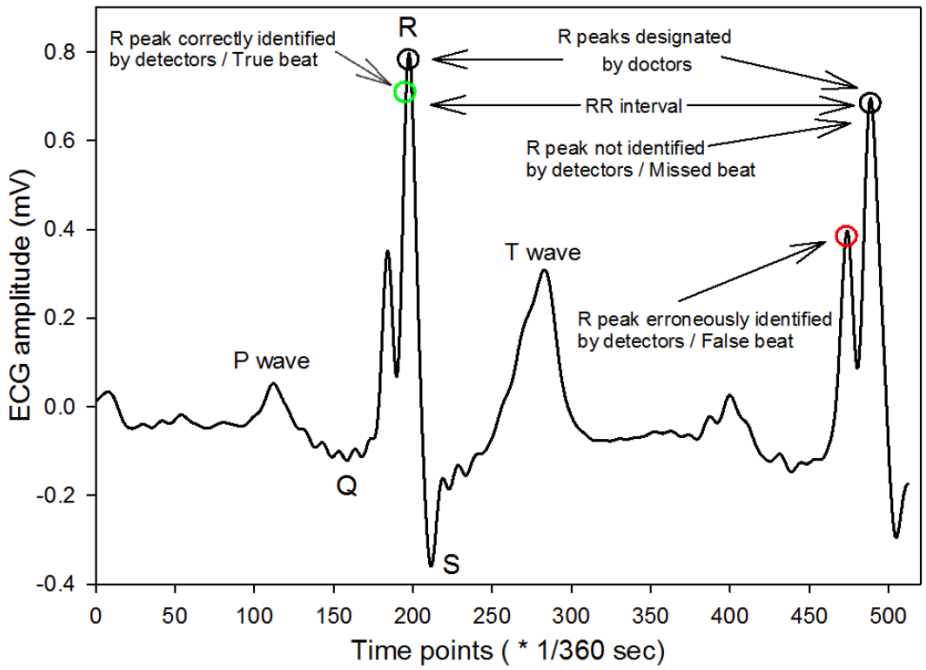


Figure 2: Ορθός και λανθασμένος εντοπισμός QRS

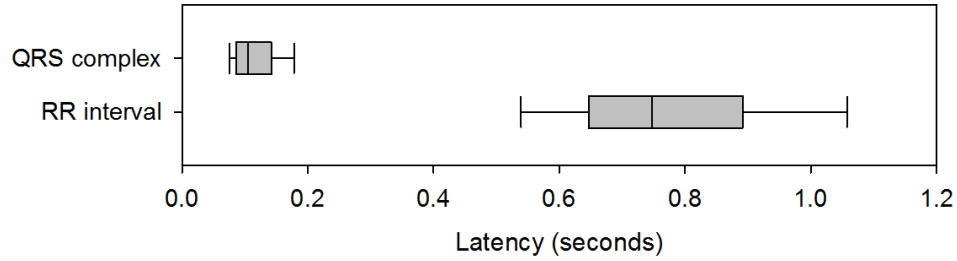


Figure 3: Χρονική διάρκεια μεταξύ δύο διαδοχικών παλμών και διάρκεια του συμπλέγματος QRS

έναν ταξινομητή μόνο με True παλμούς και του εισάγουμε False παλμούς προς ταξινόμηση. Το αποτέλεσμα ήταν ότι περίπου 86% εξ' αυτών ταξινομήθηκαν ως Abnormal. Φυσικά, οι Missed παλμοί αγνοούνται. Τελικά, παράγεται ένα σύνολο 104581 παλμών, με το διάνυσμα χαρακτηριστικών τους και τις τιμές-στόχους. Οι μισοί από τους παλμούς αυτούς χρησιμοποιούνται ως το σετ εκπαίδευσης, και οι υπόλοιποι ως το σετ επαλήθευσης.

Η διαδικασία εκπαίδευσης του ταξινομητή γίνεται offline. Μέσω design space exploration επιλέγεται το διάνυσμα χαρακτηριστικών που δίνει τον καλύτερο ταξινομητή και παράγεται το μοντέλο του ταξινομητή. Αυτό στη συνέχεια χρησιμοποιείται στο στάδιο της ταξινόμμησης για τον χαρακτηρισμό των παλμών σε real time.

Οι μετρικές που χρησιμοποιήθηκαν για το design space exploration είναι: Accuracy (number of correctly classified points/tota number of points) και Computaional cost(number of support vectros * size of feature vector). Η διαδικασία παρουσιάζεται στην εικόνα 4. Τα αποτελέσματα του design space exploration παρουσιάζονται στην εικόνα 5. Βλέπουμε πως σε όλες σχεδόν τις περιπτώσεις το accuracy ξεπερνάει το 97%. Ο καλύτερος ταξινομητής

Algorithm 1 Derived R matching

```
1:  procedure MATCHING(D_time_series, A_time_series)
2:      True_beats, False_beats, Missed_beats ← []
3:
4:      i_D, i_A ← 0
5:      len_D ← length(D_time_series)
6:      len_A ← length(A_time_series)
7:      while i_D <> len_D AND i_A <> len_A do
8:          if abs(D_times(i_R) − A_times(i_A)) <= T then
9:              True_beats ← True_beats + D_times(i_D)
10:             i_D ← i_D + 1
11:             i_A ← i_A + 1
12:         else if D_times(i_D) − A_times(i_A) > T then
13:             Missed_beats ← Missed_beats+A_times(i_A)
14:             i_A ← i_A + 1
15:         else if A_times(i_A) − D_times(i_D) > T then
16:             False_beats ← False_beats + D_times(i_D)
17:             i_D ← i_D + 1
18:         end if
19:     end while
20:
21:     return True_beats, False_beats, Missed_beats
22: end procedure
```

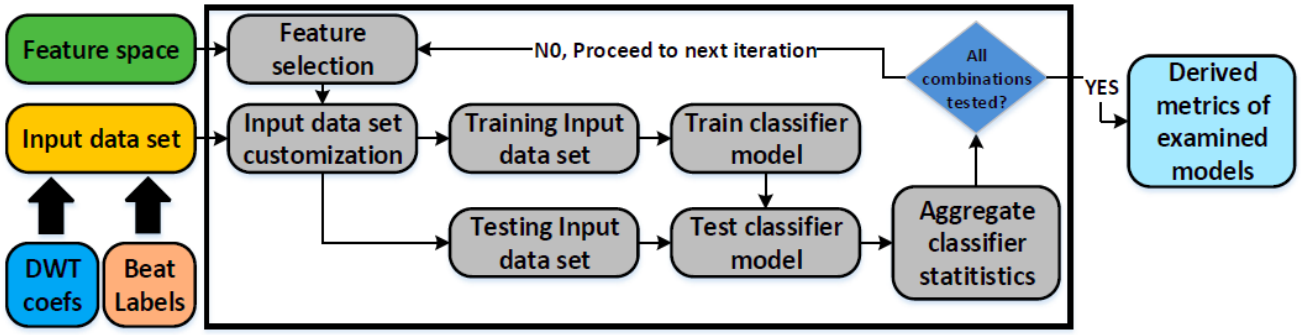
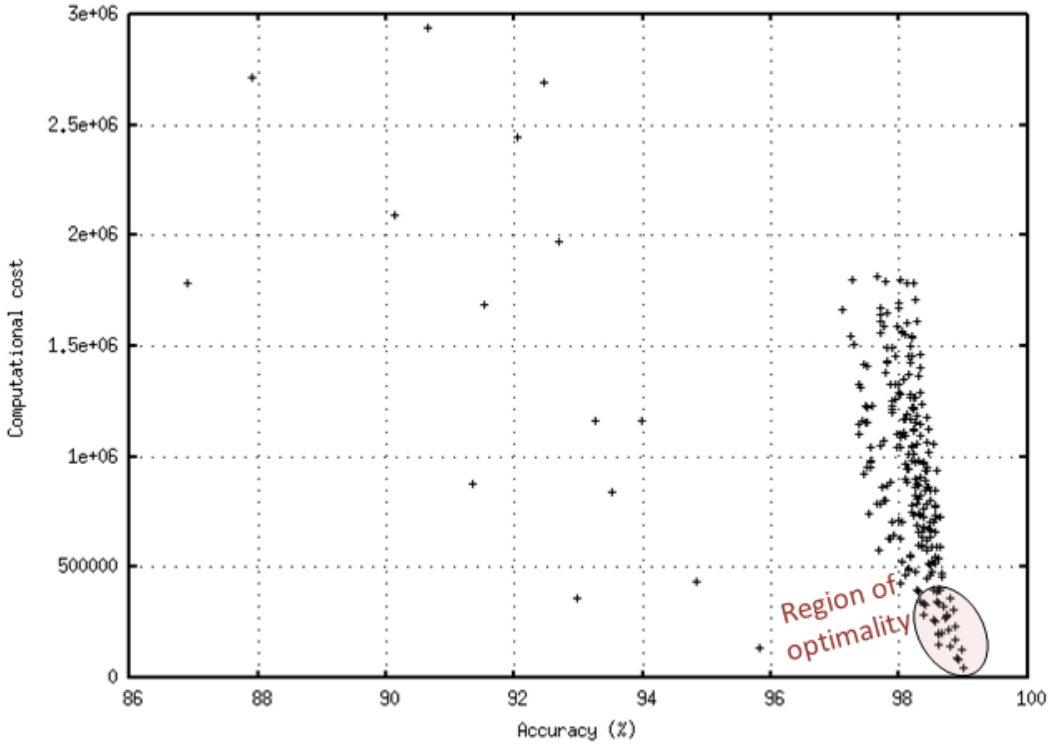


Figure 4: Design space exploration framework

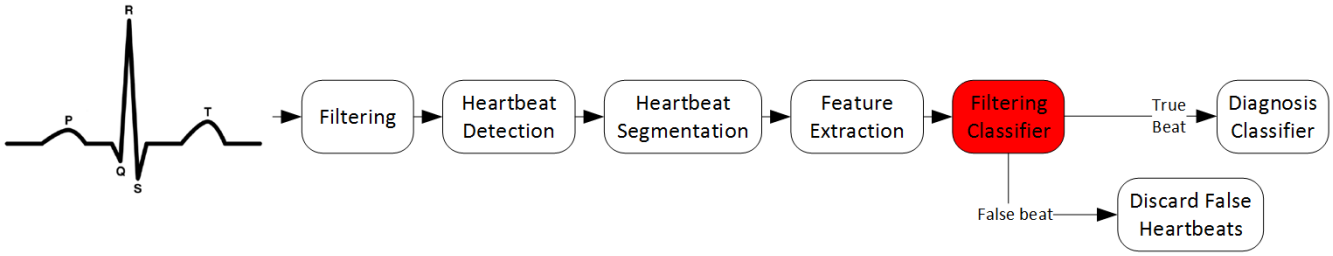
προκύπτει για διάνυσμα χαρακτηριστικών που περιέχει τα approximate coefficients του 4ου μόνο επιπέδου decomposition. Στην περίπτωση αυτή έχουμε, accuracy 98.9%, μέγεθος διανύσματος χαρακτηριστηκών 18 και 2493 support vectors. Αυτά τα αποτελέσματα δείχνουν ότι, σε σύγκριση με τα αποτελέσματα που αναφέρονται στην βιβλιογραφία, ο ταξινομητή που αναπτύχθηκε στη μελέτη αυτή παρέχει περισσότερο από ικανοποιητική απόδοση. Ωστόσο, θα πρέπει να σημειωθεί ότι λόγω των ποικιλιών στις σχετικές εργασίες στην βιβλιογραφία ως προς τις χρησιμοποιούμενες τεχνικές, η παροχή μιας απολύτως δίκαιης και αντικειμενικής σύγκρισης είναι πολύ δύσκολη.

Στο δεύτερο μέρος της εργασίας, προτείνουμε την προσθήκη ενός επιπλέον σταδίου στην αλγο-



Σχήμα 5: Design space exploration του ταξινομητή διάγνωσης

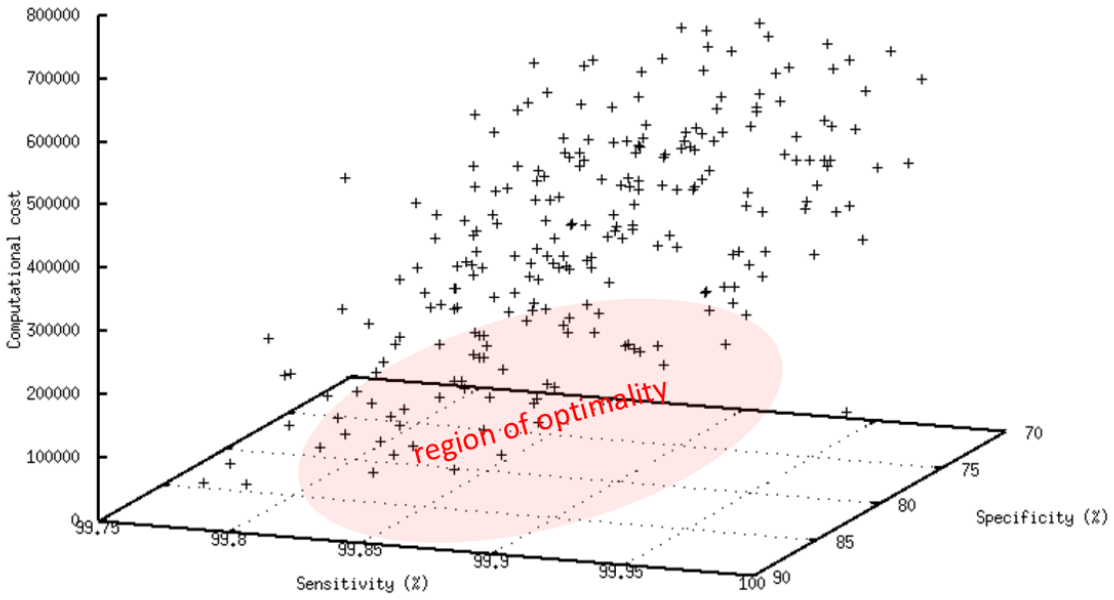
ριθμική δομή. Αυτό το στάδιο είναι τοποθετημένο ακριβώς πριν το τελικό στάδιο ταξινόμησης, και αποτελείται από ένα ταξινομητή SVM που θα λαμβάνει ως είσοδο τα χαρακτηριστικά γνωρίσματα που εξάγονται κατά το προηγούμενο στάδιο και θα ταξινομεί τους παλμούς ως αληθείς ή ψευδείς ανιχνεύσεις. Οι αληθείς παλμοί θα συνεχίζουν μέχρι το τελικό στάδιο, ενώ οι ψευδείς παλμοί θα απορρίπτονται. Στο σχήμα 6 φαίνεται η καινούργια δομή, με το κόκκινο κουτί να αποτελεί το επιπλέον στάδιο του Filtering Classifier. Για να παραχθεί το μοντέλο SVM



Σχήμα 6: Εκτεταμένη ροή ανάλυσης

του ταξινομητή φιλτραρίσματος, δημιουργούμε καινούργια σετ δεδομένων εκπαίδευσης και επαλήθευσης. Για το σύνολο των παλμών που προέκυψαν από τη βάση, χρησιμοποιούμε και πάλι ως διάνυσμα χαρακτηριστηκών τα DWT coefficients που προκύπτουν από το προηγούμενο στάδιο, ενώ οι τιμές-στόχοι τίθονται ως True ή False, βάση του αλγορίθμου που περιγράψαμε πιο πάνω. Τελικά, προκύπτουν 100231 true και 4350 false παλμοί. Οι μισοί παλμοί από κάθε κατηγορία σχηματίζουν το σετ εκπαίδευσης και οι υπόλοιποι το σετ επαλήθευσης, προκειμένου να υπάρχει ίση αναλογία των κατηγοριών στα δύο σετ. Στη συνέχεια, εφαρόζεται η ίδια διαδικασία design space exploration για την επιλογή του διανύσματος χαρακτηριστηκών και τη δημιουργία του μοντέλου SVM. Σε αυτή την περίπτωση, επικεντρωνόμαστε στις μετρικές: Sensitivity(num.

of correctly classified positive points/total number of positive points) και Specificity(num. of correctly classified negative points/total number of negative points). Τα αποτελέσματα παρουσιάζονται στο σχήμα 7. Η πιο λογική ιατρικά λύση είναι αυτή που μεγιστοποιεί το sensi-



Σχήμα 7: Design space exploration για τον προ της διάγνωσης ταξινομητή φιλτραρίσματος

tivity ώστε να ελαχιστοποιηθούν οι True παλμοί που ταξινομούνται ως False. Κατά συνέπεια, αν λάβουμε υπόψη την πλειάδα (sensitivity, specificity, computations), προκύπτει η επιλογή (99.992%, 65.56%, 272900). Ωστόσο, παρατηρούμε ότι με μικρή μείωση του sensitivity έχουμε την επιλογή (99.88%, 85.1%, 94512) με πολύ μικρότερο υπολογιστικό κόστος και αρκετά αυξημένο specificity.

Στον πίνακα 3 συγκρίνουμε τα αποτελέσματα της αρχικής και της εκτεταμένης αλγοριθμικής δομής. Μπορούμε να δούμε ότι η ακρίβεια της διάγνωσης της εκτεταμένη ροής είναι 0.01 λιγότερο σε σύγκριση με τη βασική γραμμή ροής διάγνωση. Για να το εξηγήσουμε αυτό, θα επικεντρωθεί στις άλλες μετρήσεις. Όσον αφορά την sensitivity, η εκτεταμένη ροή διάγνωση στερείται 0,05% σε σύγκριση με το το βασικό σενάριο. Αυτό συμβαίνει διότι, υπάρχει ένας μικρός αριθμός από True παλμούς οι οποίοι εσφαλμένα φιλτράρονται ως ψευδείς και ως εκ τούτου, η sensitivity της πλήρους ροής μειώνεται. Αντιθέτως, λόγω του ότι ο ταξινομητής φιλτραρίσματος απορρίπτει ένα μεγάλο αριθμό ψευδών παλμών, η specificity της εκτεταμένης ροής διάγνωσης είναι αυξημένη σε σύγκριση με το βασικό σενάριο. Το μεγαλύτερο πλεονέκτημα του ταξινομητή φιλτραρίσματος είναι ότι, αν θεωρήσουμε ότι η ροή διάγνωση προκαλεί συναγερμό κάθε φορά που ένας καρδιακός παλμός θεωρείται Abnormal, υπάρχει περίπου 75% μείωση σε αυτούς τους συναγερμούς που ανήκουν σε False beats, δηλαδή ψευδείς συναγερμούς. Το αντίτιμο είναι ότι ο ταξινομητής φιλτραρίσματος καταλαμβάνει κατά μέσο όρο 27% του χρόνου εκτέλεσης της πλήρους ροής διάγνωσης.

Στο τελευταίο μέρος της ανάλυσης, η αρχική αλγοριθμική ροή εφαρμόζεται στο IoT-based Galileo board της Intel. Για να γίνει αυτό, ο αλγόριθμος μετατρέπεται σε C κώδικα. Στην τελική του μορφή, το πρόγραμμα διαβάζει δείγμα προς δείγμα ένα ΗΚΓ ψηφιοποιημένο σε 360 δείγματα ανά δευτερόλεπτο, και η ροή ανάλυσης εκτελείται για κάθε σύνολο 3000 δειγμάτων που διαβάζεται.

Table 3: Βασική και εκτεταμένη ροή ανάλυσης

Analysis flow	Accuracy (%)	Sensitivity (%)	Specificity (%)	Number of False alarms
Baseline	97.87	99.40	99.40	2175
Extended	97.86	99.35	99.46	550

1. Filtering. Το σήμα των 3000 δειγμάτων αρχικά φιλτράρεται. Χρησιμοποιούνται τα ίδια FIR φίλτρα που χρησιμοποιήθηκαν στο Matlab.

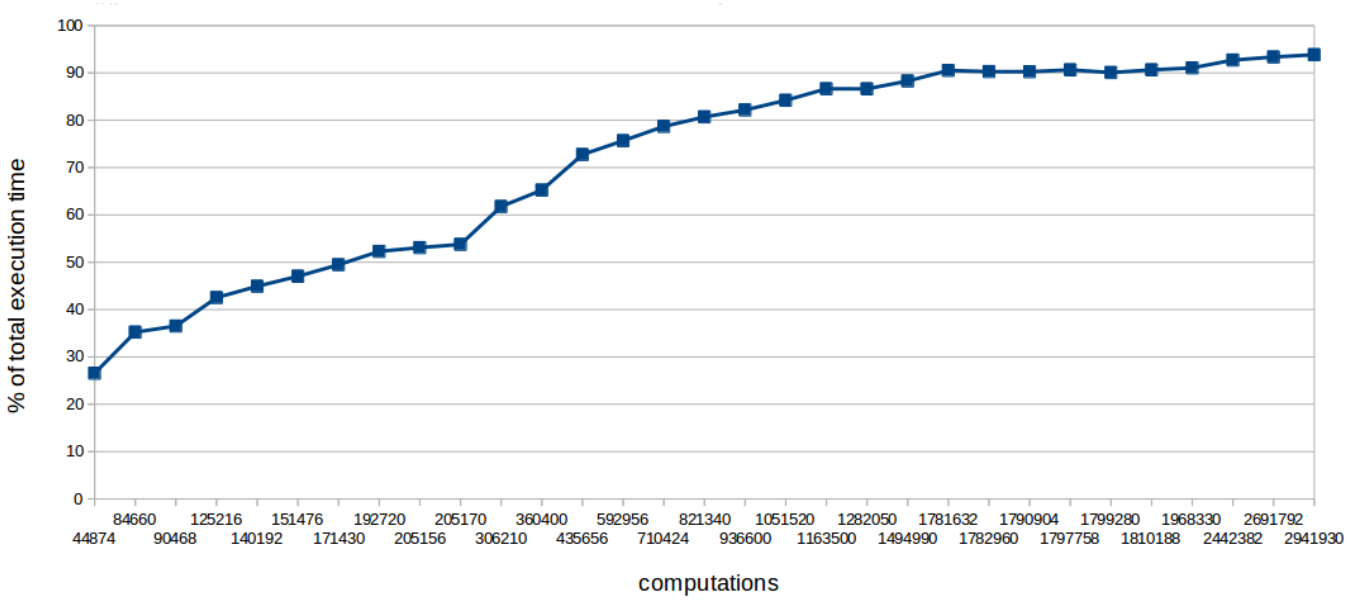
2. Heartbeat Detection. Στην υλοποίηση σε C, για το στάδιο αυτό χρησιμοποιούμε μόνο τη συνάρτηση wqrs(), καθότι μόνο για αυτή δίνεται πηγαίος κώδικας σε C. Αυτή μας δίνει τη θέση της αρχής του συμπλέγματος QRS. Αυτή τη πληροφορία, αντί να τη μεταβιβάσουμε στη συνάρτηση ecgpuwave() για να πάρουμε τη θέση της κορυφής R, τη χρησιμοποιούμε ως σημείο αναφοράς για τον παλμό, και προσαρμόζουμε το παράθυρο ανάλογα, ώστε να καλύπτει και πάλι όλο το εύρος του παλμού.

3. Heartbeat Segmentation. Το παράθυρο σε αυτή τη περίπτωση επιλέγεται ως εξής: με σημείο αναφοράς το σημείο αρχής του συμπλέγματος QRS, 86 δείγματα πριν και 170 δείγματα μετά από αυτό.

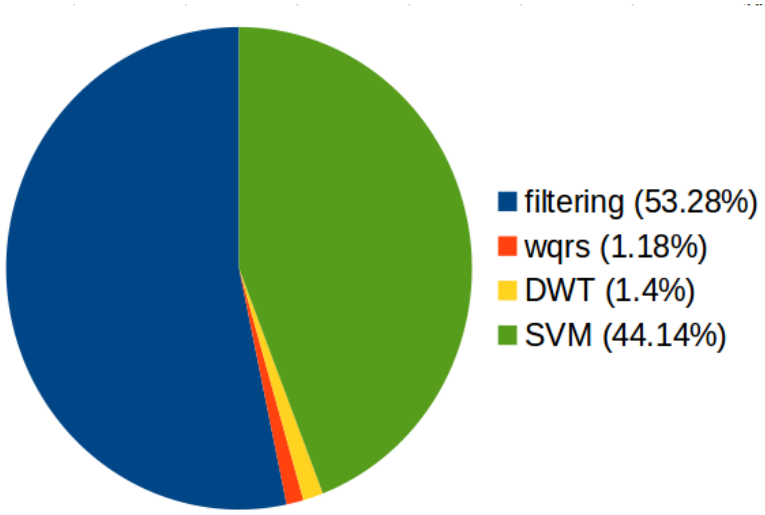
4. Feature Extractions. Για το στάδιο αυτό, υλοποιείται η συνάρτηση της Matlab wavedec() σε C. Χρησιμοποιώντας τη συνάρτηση wfilters() παίρνουμε τα φίλτρα αποσύνθεσης που σχετίζονται με το χρησιμοποιούμενο wavelet ('db2'). Ακόμα, εφαρμόζουμε symmetric-padding στο σήμα, που είναι η default ρύθμιση του Matlab για DWT. Τελικά, υλοποιούμε τη συνέλιξη του σήματος με κάθε φίλτρο.

5. Classification. Για το τελικό στάδιο της ταξινόμησης, χρησιμοποιούμε τη βιβλιοθήκη LIBSVM που περιλαμβάνει πηγαίο κώδικα της svmtrain() σε C. Μετατρέπουμε το μοντέλο SVM που παρήχθει από την ίδια συνάρτηση σε Matlab, στη μορφή που χρειάζεται η συνάρτηση σε C. Στην αρχή του προγράμματος καλείται η load_svm_model() (της LIBSVM) και φορτώνει το μοντέλο αυτό μία φορά, και στη συνέχεια χρησιμοποιείται για το στάδιο ταξινόμησης σε κάθε παλμό.

Οι 10 καλύτερες επιλογές, οι 10 πιο απαιτητικές σε υπολογιστικό κόστος, καθώς και 11 μέσες περιπτώσεις, όπως προκύπτουν από την εξερεύνηση του διαστήματος σχεδιασμού στο πρώτο μέρος της εργασίας, υλοποιήθηκαν πάνω στο Galileo. Οι ακρίβειες που επιτεύχθηκαν ήταν περισσότερο από ικανοποιητικές και το υπολογιστικό κόστος ήταν τέτοιο ώστε η ανάλυση και ταξινόμηση ΗΚΓ να μπορεί να πραγματοποιηθεί σε πραγματικό χρόνο. Στον πίνακα 4 παρουσιάζονται οι επιλογές υλοποίησης και στον πίνακα 5 τα αποτελέσματα των μετρήσεων για αυτές. Στο σχήμα 8 βλέπουμε την κλιμάκωση του χρόνου εκτέλεσης για το στάδιο ταξινόμησης, καθώς αυξάνονται οι απαιτούμενοι υπολογισμοί στα διαφορετικά μοντέλα. Στο σχήμα 9 παρουσιάζονται οι μέσοι % χρόνοι εκτέλεσης για κάθε στάδιο, από τα 10 καλύτερα μοντέλα που εξετάστηκαν. Οι αντίστοιχοι % χρόνοι εκτέλεσης για κάθε στάδιο, για 11 μέσες περιπτώσεις, φαίνονται στο σχήμα 10, και στο σχήμα 11 για τα 10 πιο βαριά υπολογιστικά μοντέλα.



Σχήμα 8: Κλιμάκωση του χρόνου εκτέλεσης σε συμφωνία με τους απαιτούμενους υπολογισμούς για κάθε διαφορετικό μοντέλο



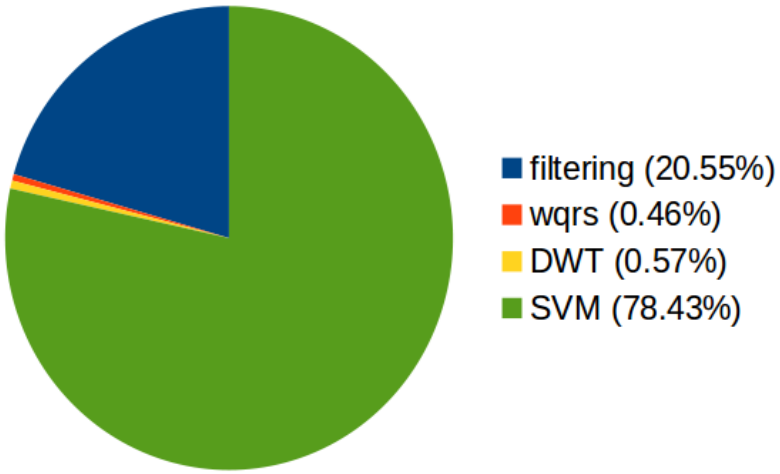
Σχήμα 9: Μέσοι % χρόνοι εκτέλεσης για κάθε στάδιο, για τα 10 καλύτερα μοντέλα

Table 4: Τα επιλεγμένα προς υλοποίηση μοντέλα του DSE

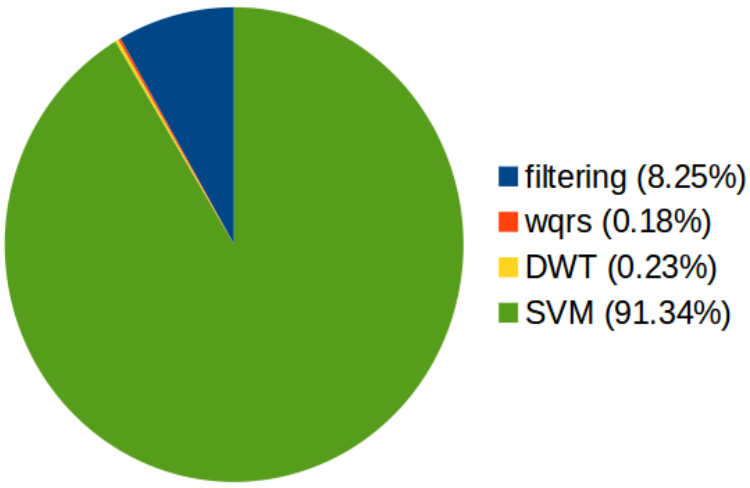
configuration	DWT coefficients	number of support vectors	feature vector size	Accuracy (%)	Computational cost
1	cA4	2493	18	98.99	44874
2	cA3	2490	34	98.93	84660
3	cA4, cD4	2513	36	98.90	90468
4	cA3, cA4	2408	52	98.99	125216
5	cA3, cD4	2696	52	98.80	140192
6	cA4, cD3	2913	52	98.62	151476
7	cA3, cA4, cD4	2449	70	98.88	171430
8	cA2	2920	66	98.62	192720
9	cA3, cD3	3017	68	98.67	205156
10	cA4, cD3, cD4	2931	70	98.62	205170
11	cA2, cA3, cA4	2595	118	98.84	306210
12	cA4, cA3, cA4, cD4	2650	136	98.79	360400
13	cD3, cD4	8378	52	94.83	435656
14	cA3, cA4, cD1	3258	182	98.36	592956
15	cA4, cD2, cD3, cD4	3861	184	98.00	710424
16	cA3, cA4, cD1, cD3, cD4	3510	234	98.26	821340
17	cA1, cA2, cA3, cA4, cD3, cD4	3122	300	98.60	936600
18	cA2, cD1, cD3, cD4	4240	248	97.70	1051520
19	cD2, cD3	11635	100	93.25	1163500
20	cA2, cA3, cD1, cD2, cD3	3885	330	98.02	1282050
21	cA1, cA3, cD1, cD2, cD4	3955	378	97.81	1494990
22	cA1, cA2, cA3, cA4, cD1, cD2, cD3, cD4	3592	496	98.22	1781632
23	cA1, cA2, cA3, cD1, cD2, cD3	3876	460	98.13	1782960
24	cA1, cA2, cD1, cD2, cD3	4204	426	97.78	1790904
25	cA1, cA2, cA3, cD1, cD2, cD3, cD4	3761	478	98.01	1797758
26	cA1, cD1, cD2, cD3, cD4	4760	378	97.28	1799280
27	cA1, cA2, cD1, cD2, cD3, cD4	4077	444	97.65	1810188
28	cD1, cD3, cD4	10815	182	92.70	1968330
29	cD1, cD2, cD4	11413	214	92.04	2442382
30	cD1, cD2, cD3, cD4	10854	248	92.45	2691792
31	cD1, cD2, cD3	12791	230	90.64	2941930

Table 5: Αποτελέσματα των υλοποιήσεων στο Intel Galileo

	filtering		beat detection		feature extraction		classification	
configuration	average (in sec)	% of total ex. time	average (in sec)	% of total ex. time	average (in sec)	% of total ex. time	average (in sec)	% of total ex. time
1	0.77127	70.01	0.01691	1.53	0.00210	1.91	0.02925	26.55
2	0.77807	61.88	0.01707	1.36	0.00192	1.53	0.04430	35.23
3	0.62330	60.49	0.01373	1.33	0.00171	1.66	0.03762	36.52
4	0.62322	54.80	0.01381	1.21	0.00166	1.46	0.04836	42.53
5	0.62386	52.48	0.01379	1.16	0.00171	1.44	0.05341	44.92
6	0.62326	50.45	0.01380	1.12	0.00173	1.40	0.05811	47.03
7	0.65056	48.14	0.01451	1.07	0.00178	1.32	0.06685	49.47
8	0.68187	45.72	0.01504	1.01	0.00147	0.99	0.07799	52.29
9	0.62327	44.76	0.01396	1.00	0.00158	1.13	0.07394	53.10
10	0.62284	44.03	0.01385	0.98	0.00173	1.22	0.07604	53.76
11	0.62648	36.43	0.01390	0.81	0.00171	0.99	0.10622	61.77
12	0.62396	33.13	0.01388	0.74	0.00171	0.91	0.12284	65.23
13	0.62414	25.97	0.01386	0.58	0.00170	0.71	0.17479	72.74
14	0.71297	23.17	0.01594	0.52	0.00199	0.65	0.23284	75.67
15	0.62899	20.30	0.01380	0.45	0.00177	0.57	0.24380	78.68
16	0.62782	18.38	0.01395	0.41	0.00176	0.51	0.27556	80.69
17	0.65934	16.99	0.01470	0.38	0.00184	0.47	0.31877	82.15
18	0.62769	15.04	0.01393	0.33	0.00175	0.42	0.35130	84.20
19	0.73301	12.75	0.01622	0.28	0.00185	0.32	0.49811	86.65
20	0.65047	12.75	0.01433	0.28	0.00173	0.34	0.44180	86.63
21	0.71573	11.15	0.01593	0.25	0.00206	0.32	0.56645	88.28
22	0.62523	09.02	0.01401	0.20	0.00181	0.26	0.62759	90.52
23	0.62374	09.28	0.01376	0.20	0.00169	0.25	0.60676	90.27
24	0.62314	09.29	0.01374	0.20	0.00171	0.25	0.60542	90.25
25	0.69495	08.90	0.01530	0.20	0.00204	0.26	0.70780	90.64
26	0.62363	09.45	0.01389	0.21	0.00179	0.27	0.59465	90.07
27	0.63113	08.91	0.01399	0.20	0.00183	0.26	0.64231	90.64
28	0.62258	08.50	0.01389	0.19	0.00177	0.24	0.66672	91.06
29	0.62295	06.95	0.01380	0.15	0.00177	0.20	0.83134	92.70
30	0.70772	06.32	0.01552	0.14	0.00199	0.18	1.04530	93.36
31	0.65894	05.89	0.01472	0.13	0.00176	0.16	1.05047	93.83



Σχήμα 10: Μέσοι % χρόνοι εκτέλεσης για κάθε στάδιο, για 11 μέσες περιπτώσεις μοντέλων



Σχήμα 11: Μέσοι % χρόνοι εκτέλεσης για κάθε στάδιο, για τα 10 πιο βαριά υπολογιστικά μοντέλα

Acknowledgments

I would like to thank Professor Dimitrios Soudris for giving me the opportunity to carry out my diploma thesis under his supervision. His advice and words of encouragement gave me motivation to improve myself.

I am extremely grateful to my supervisors, Dr. Sotiris Xydis and Vasileios Tsoutsouras, for their guidance and support. Their experience and constant assistance made it possible for me to accomplish this work. I would like to thank them for making our cooperation as productive and pleasant.

Finally, I would like to thank my family and friends for supporting me throughout my studies at the School of Electrical and Computer Engineering of NTUA.

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem Statement

Cardiovascular diseases, the leading cause of premature death in the world, are a group of diseases that involve the heart or blood vessels. They include coronary artery diseases such as angina and myocardial infarction, commonly known as a heart attack, hypertensive heart disease, rheumatic heart disease, cardiomyopathy, atrial fibrillation, congenital heart disease, endocarditis, aortic aneurysms, peripheral artery disease and venous thrombosis.
More people die annually from cardiovascular diseases than from any other cause, making it the number one cause of death globally. At the same time, as the global population pushes past 7 billion and more people reach old age, the number of deaths from cardiovascular diseases is still on the rise. According to the World Health Organization, globally, the number of deaths due to cardiovascular diseases increased by 41% between 1990 and 2013, climbing from 12.3 million deaths to 17.3 million deaths. In the United States 11% of people between 20 and 40 have CVD, while 37% between 40 and 60, 71% of people between 60 and 80, and 85% of people over 80 have CVD. These high percentages together with the increasing mortal rate, urge the need for close and continuous medical supervision and care of patients suffering from such diseases.
Meanwhile, technology scaling and improvement in electronic device manufacturing have enabled the use of medical wearable devices. Such devices can be autonomous and provide constant monitoring of patients without them being hospitalized. Since cardiovascular diseases are in the vast majority chronic ones, wearable technology can be extremely useful for the patient, covering his need for an all-day basis monitoring, while also improving his quality of life and minimizing healthcare cost by reducing the need for his hospitalization.

Another beneficial factor towards the actualization of the required medical supervision of cardiovascular disease patients, is the large-scale availability of data in the healthcare domain. Digital archives of the data generated from monitors and therapeutic devices in intensive care units and operating rooms, compose large databases. These medical data bases provided for the most part by medical research institutes, can be used for research purposes, in the effort of developing the necessary software infrastructure for medical applications. To do so, machine learning algorithms are needed in order to handle the big amounts of data. Machine learning techniques are able to examine and to extract knowledge from large databases in an automatic way, which makes them a greatly valuable tool for this type of research.

1.2 Related Work and Motivation

The investigation of the electrocardiogram (ECG) signal has been extensively used for the diagnosis of many cardiac diseases. The computer-based automatic and accurate recognition of the arrhythmias from an electrocardiographic record is essential for the correct treatment of the patient. In recent years, numerous research and various algorithms have been developed for the exertion of analyzing and classifying the ECG signal. The majority of these algorithms follow the structure of: detecting the heartbeats in an ECG signal, applying feature extraction methods on the ECG heartbeats, and using classification methods to conclude to a diagnosis.

In most of the studies, MIT-BIH database [6] is used as the source of ECG recordings. Some techniques are based on the detection of a single arrhythmia type and its discrimination from normal sinus rhythm, or the discrimination between two different types of arrhythmia. Other classes of proposed methods for arrhythmia detection and classification are based on the detection of different heart rhythms and their classification into two or three arrhythmia types and the normal sinus rhythm. Another field of interest is the ECG beat-by-beat classification, where each heartbeat is classified into one of several rhythm types.

ECG feature extraction has been studied from early time and lots of advanced techniques as well as transformations have been proposed for accurate and fast ECG feature extraction. For example, morphology and the waveform geometry [7, 8], Wavelet transform [9, 10], Hilbert transform [11], Fourier transform [12], Hermite function [13, 14], power spectral features [15], higher order spectral methods [16], nonlinear transformations such as Lyapunov exponents [17] have been used as appropriate sources for feature extraction.

The classifying methods which have been proposed during the last decade include Fuzzy Logic methods [18, 19], Artificial Neural Network [20, 10, 21], Hidden Markov Model [22], Genetic Algorithm [15], Support Vector Machines [9, 23, 24, 25], Self-Organizing Map [13], Cluster analysis [26] and other methods with each approach exhibiting its own advantages and disadvantages.

There are varieties of reported performances of automatic arrhythmia classification systems in the existing literature. As mentioned above, the methods used and the number of arrhythmia types that are classified show a great deal of variance which makes it very difficult to fairly compare the performances of different algorithms. Although there has been a tremendous amount of improvement in technology and the various approaches to the problem, automatic ECG heartbeat detection and classification with high reliability is still an open research area.

With the development of technology, wearable devices are playing an important role in ECG monitoring. Thus, the research conducted in the field of automatic ECG analysis has been, in a big part, oriented in developing efficient algorithms that overcome the constraints of embedded portable and wearable devices. Furthermore, these devices are usually intended to be part of a network, where the ECG signal is detected by wearable sensors, then sent to the wearable device where it is processed and analyzed, and finally sent to a remote device for further analysis and storage. This network of sensors, computing devices and communicator devices is referred to as the Internet of Things (IoT). In this context, the scope of this diploma thesis is to develop the software infrastructure that will support ECG signal analysis for feature extraction and the corresponding classification techniques for diagnosis of the heart condition. The implementation is done on Intel's Galileo embedded platform,

which is one of Intel's proposed IoT devices.

1.3 Summary of conducted work

In the first part of this work, we design the algorithm for ECG signal analysis and classification and implement it in Matab environment. The database chosen as source for ECG recordings is the MIT-BIH Arrhythmia database, provided by PhysioNet. The structure of the algorithm consists of the following stages: filtering, heartbeat detection, heartbeat segmentation, feature extraction, classification. The input is an ECG signal, it is filtered, the heartbeats included in it are detected, the signal is segmented into beats, features are extracted from each beat, and the output of the final stage is a label (Normal or Abnormal) for each heartbeat. For the implementation of these stages we used Matlab built-in functions, the LIBSVM library for the SVM classifier used, and also functions provided by PhysioNet in the WFDB Matlab toolkit. The classification stage consists of a SVM classifier, a supervised machine learning method. We use annotations files included in the database, which provide diagnosis labeling of each heartbeat included in each ECG recording, done by doctors. The problem detected at this point of the analysis, is that there is a mismatch in the heartbeats detected in a recording by the functions provided in the toolkit, and the heartbeats annotated by the doctors. This happens because heartbeat detectors fail to detect all heartbeats and some false detections also take place. We overcome this problem by forming a procedure that allows us to match the correctly detected heartbeats with their corresponding labels in the annotation files. Next, we perform a design space exploration over different features extracted from the signal in the feature extraction stage. These features serve as input for the classification stage. The metrics used to decided upon the best design are the accuracy and computational cost of the classification stage.

In the second part of the analysis, we suggest the addition of an extra stage to the algorithmic structure. This stage is placed right before the final classification stage, and consists of an SVM classifier that would take as input the features extracted in the previous stage and classify the heartbeats as true or false detections. True beats will continue to the final

stage, while false beats will be discarded. A design space exploration is performed similarly as done in the initial structure.

In the last part of the analysis, the initial algorithmic flow is implemented on the Intel IoT based Galileo board. To do so, the algorithm is converted in C code. In its final form, the program reads sample by sample a digitized at 360 samples per second ECG signal, and the analysis flow is executed for every set of 3000 samples that is read. The 10 best configurations, the 10 most demanding in computational cost configurations, as well as 11 configurations from inbetween, as resulted from the design space exploration in the first part of the work, were implemented on the Galileo board. The accuracies achieved were above satisfactory, and the computational cost was such so that the ECG analysis and classification can be performed in real-time.

1.4 Description of Chapters

The current thesis is structured as follows:

In chapter 2, the theoretical background of the concepts used in this thesis, is presented. Firstly, the electrocardiogram (ECG) signal is discussed in detail: the genesis of the electrical pulse from the heart, the recording of this electrical activity, and the interpretation of the ECG waveform produced. The database used as the source of the ECG recordings on which we apply the analysis that follows, is presented next: the MIT-BIH Arrhythmia Database, provided by PhysioNet. We then introduce two essential tools in signal processing and analysis, Discrete Wavelet Transform (DWT) and Support Vector Machines (SVM), that will be applied on the ECG signal during the analysis that follows.

In chapter 3, the implementation in Matlab environment of the ECG signal analysis and classification algorithm that is being proposed in this thesis, is discussed. All building blocks of the algorithmic structure are described in detail. The design alternatives are being explored, and the selection of the final configuration is made. Finally, the results of the implementation are being evaluated and compared to results of related work.

In chapter 4, we propose an extended implementation of the analysis flow presented in the

previous chapter. We address the problem this extended flow aims at, and describe the extra block added to the algorithmic structure. We, then, compare the results of the extended flow to those of the basic flow proposed in Chapter 3.

In chapter 5, the implementation of the proposed ECG signal analysis on the target embedded platform is discussed. We, firstly, present the Iot-based target platform, which is the Intel Galileo Board. Then, the implementation of the proposed algorithm in C code is described. Finally, the experimental results from the implementation of the algorithm on the Intel Galileo are demonstrated.

In chapter 6, a series of conclusions and respective observations derived from the analysis made in this thesis are discussed, and also directions for future work are pointed out.

Chapter 2

Theoretical Background

2.1 The ECG Signal

The electrocardiogram (ECG) signal has been established as one of the most fundamental bio-signals for monitoring and assessing the health status of a patient. It is produced by recording the electrical activity of the heart over a period of time using electrodes placed on a patient's body. The cardiac muscle contracts in response to electrical depolarization of the muscle cells, and the sum of this electrical activity is what the electrodes record. The ECG signal is used to examine rhythm disorders, electricity changes treatment and possible ischemia or myocardial infarction based on the intensity, duration and shape of the waveform reflecting depolarization and repolarization of the atria and ventricles.

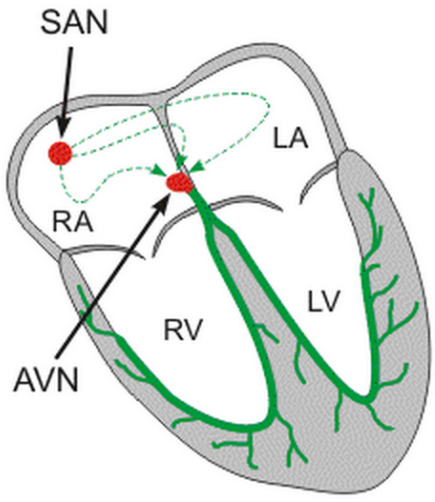


Figure 2.1: Basic physiology of the heart [1]

The heart has a specialized system that allows the genesis of rhythmic electrical pulses and the quick conduction of these pulses around the myocardium to cause effective contraction (excitation-contraction coupling). This system is susceptible to damage, particularly due to ischaemia (coronary heart disease). The electrical impulses start with the spontaneous depolarization of the sinoatrial (SA) node, an area which consists of specialized pacemaker cells, and are transmitted within the atria and ventricles through the electrical conduction system of the heart. The transmission of the electrical depolarization from atria to ventricles occurs through the atrioventricular (AV) node. The consequence of the initiation of the SA node depolarization is the depolarization of the atria. This corresponds to the P wave on the ECG waveform. The conduction through the AV node has a delay of around 150ms, which is represented by the PR interval on the ECG waveform. This delay allows the contraction of the atria while the ventricles are in a relaxation phase (increased ventricular filling, increased cardiac output), and reduces the frequency of contraction of the ventricles. Disorders in the AV node are presented as an increased PT interval, or inability to transfer the atrial depolarization to the ventricles. In rare cases there is an ectopic bundle (bundle of Kent) which allows conduction from the atria to the ventricles without delay (prestimulation), reducing the PR interval and indicating predisposition to arrhythmias (reentry mechanism). Immediately after the AV node, the electrical stimulus is conducted by high speed, through the bundle of His and the Purkinje fibers, to the myocardium of the ventricles. The depolarization of the ventricles occurs next, which is represented as the QRS complex on the ECG waveform, followed by the contraction of the ventricles. The repolarization of the ventricles is represented as the T wave on the ECG waveform. Normally the ventricles depolarize and contract almost simultaneously. Nevertheless, if the conduction is stopped at one of the bundles, then the interventricular septum and the left ventricle depolarize first, the depolarization of the right ventricle follows, and therefore the duration of the QRS complex increases and its morphology changes.

The ECG is measured by placing a series of electrodes on the surface of the body. It is possible to measure cardiac potentials that way because the body acts as a conductor of the electrical currents generated by the heart. The wave of electrical depolarization spreads from

the atria down though the interventricular septum to the ventricles. So the direction of this depolarization is usually from the superior to the inferior aspect of the heart. The direction of the wave of depolarization is normally towards the left due to the leftward orientation of the heart in the chest and the greater muscle mass of the left ventricle than the right. This overall direction of travel of the electrical depolarization through the heart is known as the electrical axis.

The fundamental principles of ECG recording are the following: A wave of depolarization heading toward the positive electrode is recorded as a positive voltage (upward deflection). A wave of depolarization heading toward the negative electrode is recorded as a negative voltage (downward deflection). A wave of repolarization heading toward the positive electrode is recorded as a negative voltage (downward deflection). A wave of repolarization heading toward the negative electrode is recorded as a positive voltage (upward deflection). A wave of depolarization or repolarization traveling perpendicular to an electrode axis results in a biphasic deflection of equal positive and negative voltages.

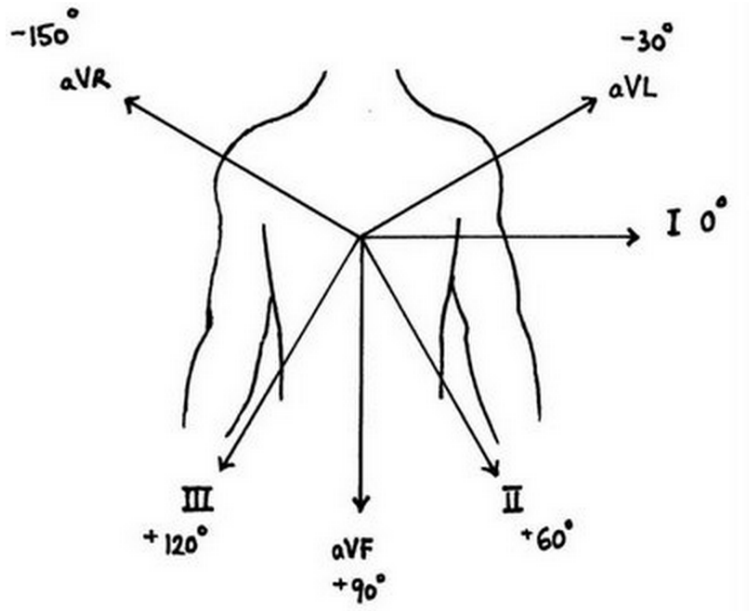


Figure 2.2: Orientation of the limb leads [2]

By convention, electrodes are placed on each arm and leg, and six electrodes are placed at defined locations on the chest. These electrode leads are connected to a device that measures potential differences between selected electrodes to produce the characteristic ECG tracings.

The limb leads record the ECG in the coronal plane, and so can be used to determine the electrical axis (which is usually measured only in the coronal plane). The limb leads are called leads I, II, III, AVR, AVL and AVF. A horizontal line through the heart and directed to the left (exactly in the direction of lead I) is conventionally labeled as the reference point of 0 degrees. The directions from which other leads 'look' at the heart are described in terms of the angle in degrees from this baseline. The chest leads record the ECG in the transverse or horizontal plane, and are called V1, V2, V3, V4, V5 and V6.

In this study, the ECG signals we examine are modified limb lead II, a bipolar lead parallel to the standard limb lead II, but acquired using electrodes placed on the torso, a requirement for long-term ECG monitoring.

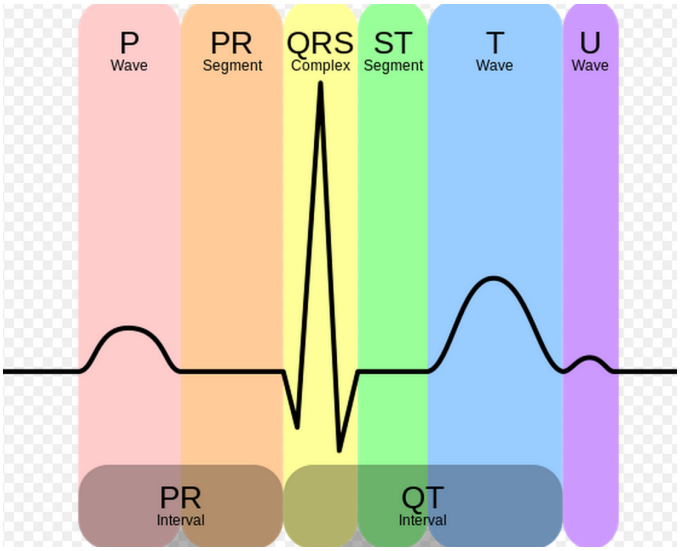


Figure 2.3: Schematic representation of normal ECG [3]

In the ECG waveform, the P wave represents the wave of depolarization that spreads from the SA node throughout the atria, and is usually 0.08 to 0.1 seconds (80-100 ms) in duration. The brief isoelectric (zero voltage) period after the P wave represents the time in which the impulse is traveling within the AV node (where the conduction velocity is greatly retarded) and the bundle of His. Atrial rate can be calculated by determining the time interval between P waves. The period of time from the onset of the P wave to the beginning of the QRS complex is termed the P-R interval, which normally ranges from 0.12 to 0.20 seconds in duration. This interval represents the time between the onset of atrial depolarization and the onset of ventricular depolarization. If the P-R interval is >0.2 sec, there is an AV

conduction block, which is also termed a first-degree heart block if the impulse is still able to be conducted into the ventricles.

The QRS complex represents ventricular depolarization. Ventricular rate can be calculated by determining the time interval between QRS complexes. The duration of the QRS complex is normally 0.06 to 0.1 seconds. This relatively short duration indicates that ventricular depolarization normally occurs very rapidly. If the QRS complex is prolonged (> 0.1 sec), conduction is impaired within the ventricles. This can occur with bundle branch blocks or whenever a ventricular foci (abnormal pacemaker site) becomes the pacemaker driving the ventricle. Such an ectopic foci nearly always results in impulses being conducted over slower pathways within the heart, thereby increasing the time for depolarization and the duration of the QRS complex. The shape of the QRS complex in the above figure is idealized. In fact, the shape changes depending on which recording electrodes are being used. The shape will also change when there is abnormal conduction of electrical impulses within the ventricles.

The isoelectric period (ST segment) following the QRS is the time at which the entire ventricle is depolarized and roughly corresponds to the plateau phase of the ventricular action potential. The ST segment is important in the diagnosis of ventricular ischemia or hypoxia because under those conditions, the ST segment can become either depressed or elevated.

T wave.

The T wave represents ventricular repolarization and is longer in duration than depolarization (i.e., conduction of the repolarization wave is slower than the wave of depolarization). Sometimes a small positive U wave may be seen following the T wave. This wave represents the last remnants of ventricular repolarization. Inverted or prominent U waves indicates underlying pathology or conditions affecting repolarization.

The Q-T interval represents the time for both ventricular depolarization and repolarization to occur, and therefore roughly estimates the duration of an average ventricular action potential. This interval can range from 0.2 to 0.4 seconds depending upon heart rate. At high heart rates, ventricular action potentials shorten in duration, which decreases the Q-T interval. Because prolonged Q-T intervals can be diagnostic for susceptibility to certain types of tachyarrhythmias, it is important to determine if a given Q-T interval is excessively long.

In practice, the Q-T interval is expressed as a "corrected Q-T (QTc)" by taking the Q-T interval and dividing it by the square root of the R-R interval (interval between ventricular depolarizations). This allows an assessment of the Q-T interval that is independent of heart rate. Normal corrected Q-Tc intervals are less than 0.44 seconds. There is no distinctly visible wave representing atrial repolarization in the ECG because it occurs during ventricular depolarization. Because the wave of atrial repolarization is relatively small in amplitude, it is masked by the much larger ventricular-generated QRS complex.

2.2 MIT-BIH Database

As already stated, for the purposes of this study, data from the MIT-BIH arrhythmia database [27] were used. This database is a result of the collaboration of Beth Israel Deaconess Medical Center and MIT, and it is one of the most utilized databases for research purposes.

The database is composed of 48 half-hour excerpts of two-channel (two leads) ambulatory ECG recordings, obtained from 47 subjects. Of these, twenty three recordings were chosen at random from a collection of over 4000 24-hour ambulatory ECG recordings, serving as a representative sample of routine clinical recordings. The remaining twenty five recordings were selected from the same set to include a variety of rare but clinically important phenomena (complex ventricular, junctional and supraventricular arrhythmias), which would not be well represented in a small random sample. The subjects included 25 men aged 32 to 89 years and 22 women aged 23 to 89 years. Approximately 60% of the subjects were inpatients and 40% outpatients. The data are bandpass filtered at 0.1-100Hz and digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. In 45 recordings, the first lead is a modified limb lead II (MLII), and for the resting 3 recordings it is lead V5. The second lead is lead V1 for 40 of the recordings, and it is either lead II, V2, V4 or V5 for the other recordings.

The MIT-BIH database also provides annotations for each record, where cardiologists placed a label for every beat detected in the record. There are approximately 110.000 annotations.

Table 2.1 lists the heartbeat types included in the database and their mapping to the American Heart Association (AHA) heartbeat classes (N, V, F, E, P, Q and O). Table 2.2 shows the percentages of beat labels that correspond to each heartbeat class. The percentages are disproportionate, as the largest beat class, 'N' (normal beat), covers 84,8% of the beats found in the database.

In the algorithmic analysis that follows, data from the first-channel lead (MLII) of all records of the database were used, apart from records 102, 104, 114, whose first-channel lead is not a MLII. Two arrhythmia groups are examined, 'Normal' (N), and 'Abnormal' (V, F, E, P, Q, O).

Table 2.1: Mapping the MIT-BIH heartbet types to the AHA heartbeat classes

AHA heartbeat class	**N**	**V**	**F**	**E**	**P**	**Q**	**O**
MIT-BIH heartbeat types	**N (normal)**	**V (premature ventricular contraction)**	**F (fusion of ventricular and normal)**	**E (ventricular escape)**	**P (paced)**	**Q (unclassifiable)**	**! (ventricular flutter wave)**
	L (left bundle branch block)		**f (fusion of paced and normal)**				**P (non-conducted P wave)**
	R (right bundle branch block)						
	A (atrial premature)						
	a (aberrated atrial premature)						
	J (nodal premature)						
	S (supraventricular contraction)						
	e (atrial escape)						
	j (nordal escape)						

Table 2.2: Beats of full database corresponding to each class

Heartbeat class	N	V	F	E	P	Q	O	total
number of beats	93411	7129	1785	106	7028	33	665	110157
% of total beats	84.8	6.47	1.62	0.096	6.38	0.03	0.6	100

2.3 Discrete Wavelet Transform

The Wavelet Transform (WT) is similar to the Fourier transform, with the extension that it is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal. This is essential when analyzing non-stationary signals (whose frequency response varies in time), such as the ECG signal, where the time localization of the frequency spectral components are needed. Generally, the wavelet transform can be expressed by the following equation:

$$F(\alpha, b) = \int_{-\inf}^{\inf} f(x)\psi^*_{(\alpha,b)}(x)dx$$

where the * is the complex symbol and function ϕ is the transforming function called the mother wavelet. Dilation, also known as scaling, compresses or stretches the mother wavelet and translation shifts it along the time axis. The WT can be categorized into continuous and discrete. Continuous, in the context of the WT, implies that the scaling and translation parameters change continuously. However, calculating wavelet coefficients for every possible scale can represent a considerable effort and result in a vast amount of data. Therefore, discrete wavelet transform (DWT) is often used.

In the DWT, a time-scale representation of a digital signal is obtained using filtering techniques. Filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the

signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling operations. Downsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. Upsampling a signal corresponds to increasing the sampling rate by adding new samples to it (usually zeros or interpolated values).

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and high pass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal x[n] is first passed through a halfband highpass filter g[n] and a lowpass filter h[n]. After filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p/2 radians instead of p. The signal can therefore be downsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_{n} x[n] * g[2k - n]$$

$$y_{low}[k] = \sum_{n} *h[2k - n]$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after downsampling by 2.

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure can be repeated for further decomposition.

2.4 Support Vector Machines

Support vector machines (SVMs) are machine learning algorithms, based on the statistical learning theory, that analyze data and recognize patterns. They are used for classification and regression analysis. Given a set of training examples, each labeled for belonging to one of two categories (supervised learning), an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. SVMs can only handle binary classification problems. Multiclass classification can be obtained through the combination of multiple binary classifiers.

An essential component of SVMs is the separating hyperplane. In a binary classification task, the hyperplane is the geometrical division or separation between the two categories. In a one- dimensional space, this is a single point, in a two-dimensional space a line, in a three-dimensional space a plane. We can extrapolate this procedure mathematically to higher dimensions. The general term for a separator in such a high dimensional space is a hyperplane. The SVM algorithm will try to find the optimal hyperplane, called maximum margin hyperplane that offers the best classification. This is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier. The decision of the optimal hyperplane is fully specified by a (usually small) subset of the data which defines the position of the separator. These points are referred to as the support vectors. In order for the SVM to be able to deal with errors in the data by allowing a few misclassification, soft margins can be set around the hyperplane. They determine the number of examples that are allowed to push their way through the margin of the hyperplane at a certain distance without affecting the final result.

A SVM is a kernel-based technique that makes use of a kernel function. While the original problem may be stated in a finite dimensional space, it often happens that the categories to discriminate are not linearly separable in that space. For this reason, it is proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, making the separation easier in that space. A kernel function will add a dimension to data, in order to obtain the most optimal classification. Any given dataset with consistent labels can be

brought into a dimension where it can be linearly separated by a hyperplane.

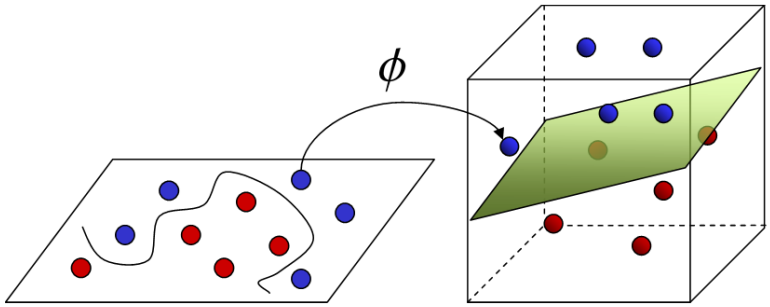


Figure 2.4: Separation of data classes possible in a higher-dimensional space

The set of instance-label pairs $(x_i, y_i), \quad i = 1, ..., l$ where $x_i \in R^n$ and $y \in \{1, -1\}^l$, are said to be linearly separable if there exists a vector w and a scalar b such that:

$$y_i(w^T x_i + b) \geq 1, \quad i = 1, ..., l.$$

The optimal hyperplane:

$$w_0^T x + b_0 = 0$$

is the unique one which separates the input data with a maximal margin.

To allow classification with an error term, we introduce some non-negative variables $\xi_i \geq 0$, $i = 1, ..., l$.

Finally, the SVM requires the solution of the following optimization problem:

$$\begin{aligned} min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ subject \quad to \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned}$$

where C is the penalty parameter of the error term.

By introducing Lagrange multipliers, α_I, the solution is given by:

$$w = \sum_{i=1}^{l} y_i \alpha_i x_i$$

Only a small fraction of the α_i coefficients are non-zero. The corresponding pairs of x_i entries are the support vectors and they fully define the decision boundary.

The hyperplane decision function can be written as follows:

$$f(x) = sgn[\sum_{i=1}^{l} y_i \alpha_i (K(x, x_i) + b]$$

where $K(x, x_i)$ is the kernel function with which the input data are mapped to a higher dimensional space.

Some typical examples of kernel functions used are the following:

- linear: $K(x_i, x_j) = x_i^T x_j$
- polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- radial basis function(RBF): $K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0$
- sigmoid: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$

Chapter 3

Design and Exploration of ECG Analysis Flow

3.1 Algorithmic Structure

Along with the MIT-BIH annotated ECG records, a library of software for physiologic signal processing and analysis is provided by PhysioNet [28]. The initial stage of this study was implemented in Matlab environment, and so the WFDB Toolbox for Matlab was used for the reading and processing of the database.
More specifically, the functions that were used, are:

- *rdsamp*, to read a record and obtain a vector of the sample amplitudes for each signal (two leads) contained in the record, and a vector representing the sampling intervals
- *rdann*, to read the annotation file of a record and obtain a vector with the sample numbers where labels have been placed, as well as the type of the labels
- *wqrs*, to find the QRS complexes of the beats contained in an ECG signal record
- *ecgpuwave*, to find the peaks of each wave of the beats contained in an ECG signal record

Figure 3.1 illustrates the structure of a typical analysis algorithm for heartbeat classification, which is the one that was implemented in this study. The first lead of the digitized ECG signal is applied as the input to the system. A filtering unit is used as a preprocessing stage, to remove baseline wander and noise from the ECG signal. The filtered signal is then passed to the heartbeat detection unit, which attempts to locate all the heartbeats

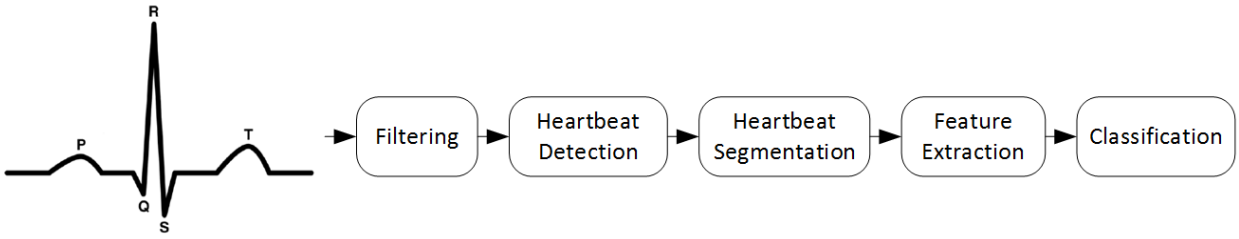


Figure 3.1: Proposed ECG analysis flow

contained in the input ECG signal. Next in the flow, is the segmentation unit, where the input signal is segmented into single heartbeats, accordingly with the information extracted from the previous stage. In order to achieve greater classification performance, a feature extraction unit is included. There, for each produced heartbeat, a feature vector is extracted, containing a smaller number of elements than the ECG samples forming the heartbeat. This vector serves as input for the classification stage, where the heartbeat is labeled as either 'N' (normal) or 'ABN' (abnormal) by a single classifier.

The stages in more detail:

1. Filtering

   All ECG records were filtered with a band-pass filter, in order to remove noise [29]. ECG noise can be classified into the following categories:

   - Power line interference
   - Electrode pop or contact noise
   - Patient-electrode motion artifacts
   - Electromyographic (EMG) noise
   - Baseline wandering

   Among these, the power line interference and the baseline wandering are the most significant and they can strongly affect ECG signal analysis. The Matlab function $filter()$ was used to filter the signals with a band-pass filter of bandwidth 1Hz to 50Hz. The filtered ECG signals were used in the rest of the processing.

2. Heartbeat detection

   Firstly, the wqrs function is applied to the signal, which gives us the locations of all QRS complexes found in the signal. This information along with the ECG signal, are the inputs to ecgpuwave function, which gives us the exact position of all the R peaks found in the signal. QRS detection, especially detection of R wave in heart signal, is easier than other portions of ECG signal due to its structural form and high amplitude. Each R peak detection corresponds to the detection of a single heartbeat. The ecgpuwave program by Laguna el al. has been validated on the Common Standards in Electrocardiography Multilead database [30] and the MIT-BIH QT database [31] and its performance in detecting significant points in ECG signals is comparable to those given by experts.

3. Heartbeat segmentation

   Having located the R peaks of each heartbeat waveform, we can proceed to segment the ECG signal into single heartbeats. To do that, we have to decide on a window width, which having as center the detected position of the R peak, will cover the whole of the heartbeat waveform. We choose a window width of 257 samples, as suggested in [9].

4. Feature extraction

   As a feature extraction mechanism we use Discrete Wavelet Transform (DWT) [32], since it has been proven to produce very accurate results. The wavelet base for the DWT is Daubechies of order 2 (db2) and we perform 4 levels of decomposition as proposed in [9]. The Matlab function $wavedec()$ was used to perform the wavelet analysis on the signal, and functions $appcoef()$ and $detcoef()$ were used to extract the approximation and detail coefficients, respectively. The DWT is used to compute compressed parameters of the heartbeat data which are called features. These parameters characterize the behavior of the heartbeat. The method of using a smaller number of parameters to represent the heartbeat is particularly important for recognition and diagnostic purposes. The 4 levels of decomposition produce 8 sets of coefficients each one for 4 levels of detailed and 4 levels of approximate coefficients. Since the heart-

beat on which the DWT is applied consists of 257 samples, the number of wavelet coefficients for the first, second, third and fourth level, are respectively 130, 66, 34 and 18. Thus, 494 wavelet coefficients are obtained for each heartbeat. The final feature vector that serves as input to the classification stage, resulted from a design space exploration performed on all combinations of these 8 sets of coefficients, as will be discussed later on.

5. Classification

The last stage of the structure, consists of a binary classifier, which labels each heartbeat as either 'Normal' or 'Abnormal'. In this study, we focus on using a Support Vector Machine (SVM) classifier [33], mainly due to its ability to support non-linear classification with efficient accuracy and computation cost. The Matlab interface of LIBSVM [34], a library for Support Vector Machines, was used for the implementation of the SVM classifier.

A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one "target value" (the heartbeat class label – 'Normal' or 'Abnormal' – acquired from the annotations) and several "attributes" (the feature vector produced from the previous stage). The goal of the SVM is to produce a model which, based on the training data, predicts the target values of the test data given only the test data attributes.

As kernel function, we use the radial basis function (RBF):

$$K(x_i, x_j) = exp(-\gamma||x_i – x_j||^2), \gamma > 0$$

where γ is a kernel parameter. The RBF is a reasonable choice as kernel function. This kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear, with smaller complexity in the model selection than the polynomial kernel does.

There are two parameters, C and γ, which need to be tuned, so that the model which

best predicts the unknown data is selected. Parameter γ is set to $1/k$, where k means the number of attributes in the input data, and parameter C is set to 1, which is the default option recommended by LIBSVM.

The training process of the SVM model can be done offline. Having a training data set and a testing data set for evaluation, we perform the design space exploration that is described below to decide upon the feature vector which gives the model with the best performance results (different feature sets generate different SVM models). Then, with a fixed feature vector, the SVM model is produced offline. This model is used for classification of heartbeats inputted in the algorithmic flow, in real time. The complete algorithmic analysis including the offline training and the online heartbeat classification is shown in Fig. 3.2.

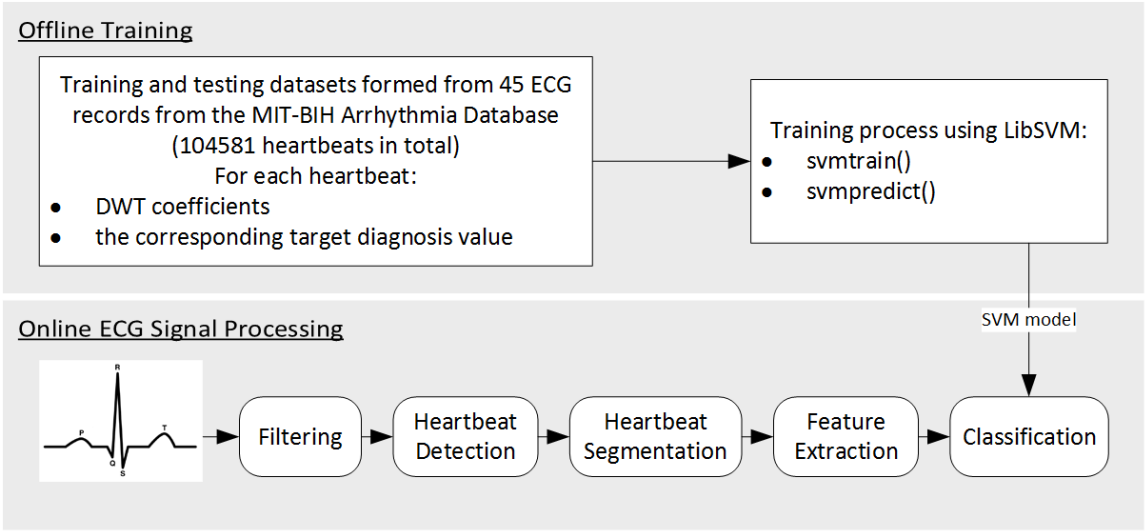


Figure 3.2: Offline training and online classification.

3.2 Creating Classification Datasets

In order for the SVM model to be produced, we need to form the training data set and the testing data set, using the 45 selected ECG records from the MIT-BIH database. As the previous flow indicates, all records were firstly filtered, the R-peaks were located, and then the records were segmented into single heartbeats, forming a set of 104581 heartbeats. As stated before, for each heartbeat we need a vector of attributes and a target value.

The vector of attributes is the feature vector which contains the DWT coefficients of the heartbeat and is formed at the feature extraction stage. As target value, we use the label given for each heartbeat in the annotation files. In reality, QRS detectors fail to correctly identify all QRS waves contained in the input signal. An ECG signal in the form of a time series is provided as input to the detector, which generates a set of points in time corresponding to the R peaks. However comparing the detected points with the respective annotations, there is a mismatch not only in the number of identified R peaks but also in their exact time value. In Fig. 3.3, black circles indicate the R peaks defined by medical experts. The rest of the circles indicate points as they have been identified by the R peak detector used. According to the proximity of these points compared to the annotated R peaks, the detected R peaks can be classified as (i) True, (ii) False or (iii) Missed. A True R peak is one close to an R peak annotation (green circle in Fig. 3.3). A False R peak is one which is far from the corresponding R peak annotation and is erroneously identified by the detector (red circle in Fig. 3.3). In case that the detector has failed to identify the second R peak of Fig. 3.3, it is considered to be a missed one.

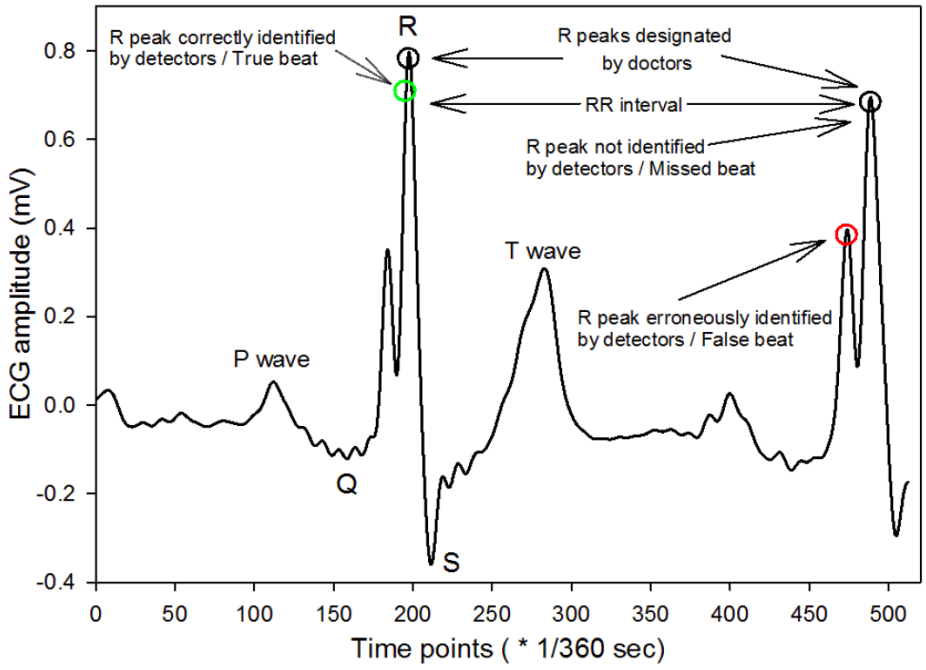


Figure 3.3: Correct vs. faulty QRS detection [4]

In order to derive the class of a detected R peak, we use the distance from the actual R

peak. Thus, there is the need to define a threshold T to characterize the heart-beat class. A statistical analysis of the ECG signals provided by the MIT-BIH database was conducted and focused on deriving a robust estimation about the time lapse between two consecutive heart beats and the duration of the QRS complex. The results regarding these durations are summarized in fig. 3.4. The values are in seconds and they are measured using the entire MIT-BIH database and the corresponding annotations. It is evident that the value which exhibits the most robust behavior is the QRS complex duration and therefore its median value which is approximately 0.11 seconds will be used to determine the threshold T. With an ECG's sampling frequency Fs = 360Hz, QRS complex is approximately 40 data samples wide. We define the threshold T as half the duration of the QRS complex, ergo 20 samples. Having defined the threshold T, a rule based procedure was realized to quantify

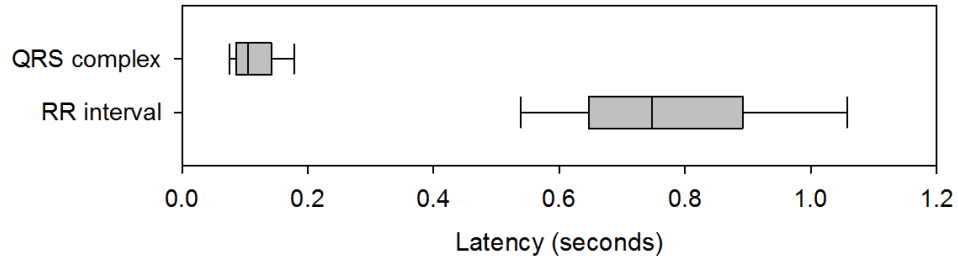


Figure 3.4: Latency values of heartbeats

the detected R peak by a detector as (i) True, (ii) False or (iii) Missed, using as reference input the respective annotated R peaks. The two discrete time series, i.e. D: detected and A: annotated R peaks, are traversed in parallel and an index in each one (i_D and i_A, respectively) indicates which R peaks are compared. If the detected and annotated R peaks under examination are at most T time units away, $|D[i_D] - A[i_A]| \leq T$, then the detected is considered True and both indexes are increased. If the detected R peak follows after more than T points, $D[i_D] - A[i_A] > T$, then the annotated R peak is considered a missed one and i_A is increased. Conversely, if the annotated R peak precedes for more than T points, $A[i_A] - D[i_D] > T$, the detected one is considered False and i_D is increased. This process takes place at the ECG analysis flow at design time and in no case are doctor annotated files utilized in ECG analysis flow at run-time.

Algorithm 2 Derived R matching

```
1: procedure MATCHING(D_time_series, A_time_series)
2:     True_beats, False_beats, Missed_beats ← []
3:
4:     i_D, i_A ← 0
5:     len_D ← length(D_time_series)
6:     len_A ← length(A_time_series)
7:     while i_D <> len_D AND i_A <> len_A do
8:         if abs(D_times(i_R) − A_times(i_A)) <= T then
9:             True_beats ← True_beats + D_times(i_D)
10:            i_D ← i_D + 1
11:            i_A ← i_A + 1
12:        else if D_times(i_D) − A_times(i_A) > T then
13:            Missed_beats ← Missed_beats+A_times(i_A)
14:            i_A ← i_A + 1
15:        else if A_times(i_A) − D_times(i_D) > T then
16:            False_beats ← False_beats + D_times(i_D)
17:            i_D ← i_D + 1
18:        end if
19:    end while
20:
21:    return True_beats, False_beats, Missed_beats
22: end procedure
```

This procedure allows us to match the true detected heartbeats with their corresponding labels in the annotation files. As for the falsely detected heartbeats, we set the target value as 'Abnormal'. This seems as a logical assumption, but to validate this choice, we train a classifier with only true detected heartbeats and evaluate it under false heartbeats testing scenarios. The result was that about 86% of the heartbeats were indeed classified as Abnormal. Of course, the missed heartbeats are not taken into consideration, since they are not detected during the heartbeat detection stage.

Finally, we have a data set of 104581 heartbeats (100231 true heartbeats and 4350 false heartbeats), along with a vector of their corresponding target values, and a matrix with the DWT coefficients for each heartbeat. Half of the heartbeats are used as the training data set for the SVM model, and the rest of the heartbeats are used as the testing data set.

3.3 Design Space exploration for SVM tuning

The design goal is to create an SVM based classifier which given a heartbeat to be classified in a binary manner aims at:

- Sensitivity = $\frac{\text{Num. of correctly classified Positive points}}{\text{Total number of Positive points}}$
  to be maximized i.e. the best possible recognition of Normal heartbeats is achieved.
- Specificity = $\frac{\text{Num. of correctly classified Negative points}}{\text{Total number of Negative points}}$
  to be maximized i.e. the best possible recognition of Abnormal heartbeats is achieved.
- Accuracy = $\frac{\text{Number of correctly classified points}}{\text{Total number of points}}$
  to be maximized
- Minimizing the computational cost

To achieve these design goals, we performed a design space exploration on the feature vectors produced by the DWT feature extraction phase. As already discussed, the 4 levels of decomposition performed on the input heartbeat, produce 8 sets of coefficients, 4 sets of detailed coefficients (dc1, dc2, dc3, dc4) and 4 sets of approximate coefficients (ac1, ac2, ac3, ac4). As a feature vector we use combinations of these 8 sets. The implemented exploration framework is illustrated in Fig. 3.5. To reduce exploration execution we input to it the data set accompanied by the extracted sets of DWT coefficients for each heartbeat. DWT is calculated only once, thus saving a great amount of time.

From that point on, there is an iteration over the combinations of coefficients that the designer would like to evaluate as a feature vector for the description of heartbeat. In more detail we refer to combinations of coefficients as picking any of the 8 sets of coefficients and testing them as the selected features for the building of the classifier. Each combination results in a different classifier both in accuracy and number of support vectors. The product of number of support vectors and size of feature vector is a reliable metric for inducing the computational requirements of the produced classifier since it is the number of multiplications required for a new heart beat to be classified [33].

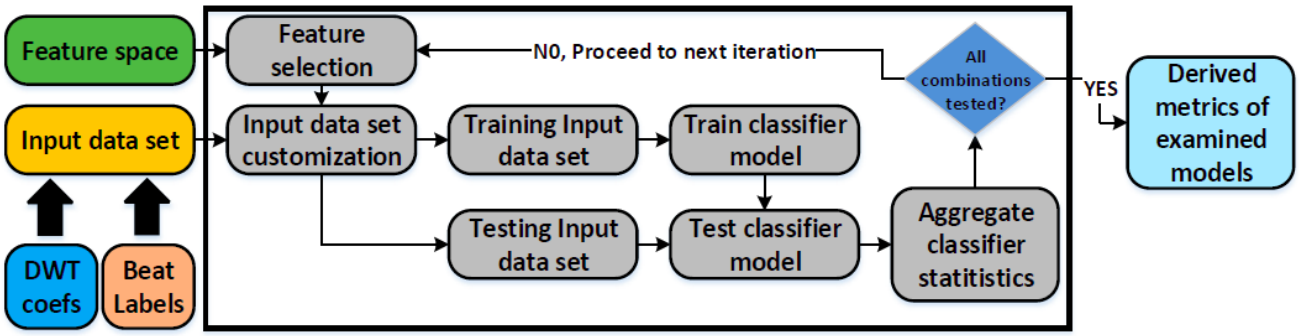


Figure 3.5: Design space exploration framework

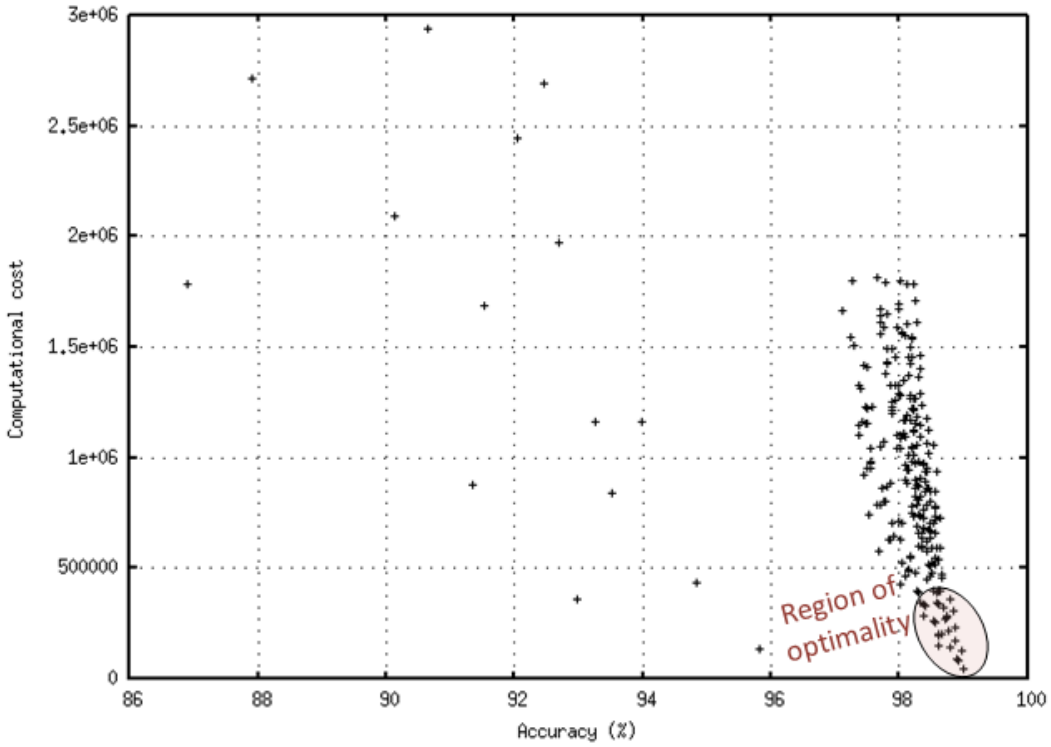


Figure 3.6: Design space exploration of diagnosis classifier

Figure 3.6 presents the results of the design space exploration, regarding accuracy and computational cost. As we can see, in almost all cases the accuracy is above 97%, with only a few exceptions. The best result comes of the feature vector which only contains the approximate coefficients of the 4th level of decomposition. In this case, we have an accuracy of 98.9%, a feature vector of size 18, and 2493 support vectors.

3.4 Result Evaluation

Table 3.1: Result comparison

Authors	Classes	Feature set	Classifier	Performance measures
This study	2 beat groups	4-level DWT	SVM	98.9% ACC
de Chazal et al. [7]	5 beat groups	R-R interval, heartbeat intervals, morphology	Linear Discriminant	73% SEN, 96% SPE
Yeh et al. [26]	5 beats types	QRS-dur, QTP-int, ratio-RR, area-R'ST'	Cluster analysis	94.3% ACC
Übeyli [9]	4 beat types	Statistics of 4-level DWT	SVM	98.61% ACC
Güler and Übeyli [20]	4 beat types	Hos features of DWT Coefficient	CNN	96.94% ACC
Lagerhlom et al. [13]	4 beat groups	Hermite functions, RR-interval	SOM	82.2% SEN, 98.9% SPE
Asl el al. [24]	6 beat types	Linear analysis features, nonlinear analysis features, feature reduction by GDA	SVM	99.16% ACC

The comparison of the system constructed in this study with similar systems in the literature is a difficult task due to varieties in the classification techniques, number of classes, data sources, and measures used for reporting performance results. Nevertheless, some conclusions can be drawn.

These results show that, compared to reported results in the literature, the classifier designed in this study provides above satisfactory performance. However, it should be noted due to the varieties in the related works in the literature as mentioned previously, providing a completely fair and objective comparison is very difficult.

Chapter 4

Extended ECG Analysis Flow for Faulty Heartbeat Detection

4.1 Problem Addressed: Faulty Heartbeat Detection

As described previously, QRS detectors fail to correctly identify all points of interest. We evaluated the accuracy of the three detection algorithms found in the Physionet tool-suite: (i) WQRS, (ii) SQRS, (iii) GQRS. In Table 4.1, it is shown that the false and missed heartbeats are significantly lower than the true detected heartbeats, but still a considerable number. In an effort to create a successful ECG analysis flow this erroneous behavior should be minimized.

In addition, we evaluated the typical ECG analysis flow proposed in the previous chapter under false heartbeats testing scenarios. In that flow the classifier is only trained to determine Normal or Abnormal heartbeats using as reference the annotated R peaks. We reported that about 86% of the false heartbeats were classified as Abnormal. This shows that faulty detected heartbeats severely affect the diagnostic abilities offered by the ECG analysis flow, as most of them create 'false alarms'.

Table 4.1: Beat categories for different detectors

Detector	False beats	Missed beats	True beats
wqrs	3823	3571	100293
sqrs	3231	4312	99552
gqrs	3473	3419	100445

4.2 Extended Algorithmic Structure

As a solution to this problem, we propose a machine-learning based classifier that extends the typical ECG analysis flow with the responsibility of distinguishing whether an R peak determined by one of the aforementioned detectors is True or False.

The extended structure is shown in Fig. 4.1. The red box indicates the extra stage of Filtering, where a classifier characterizes the beat detected on the respective previous stage, either as True and passes it on to the next stage, or as False and discards it. All other stages are the same as the ones described in the typical ECG analysis flow. The filtering stage is placed after the feature extraction stage. This is very important since this way the filtering classifier is based on the same features extracted for the diagnosis classifier (of the final stage), meaning that it does not burden the flow with an extra feature extraction process.

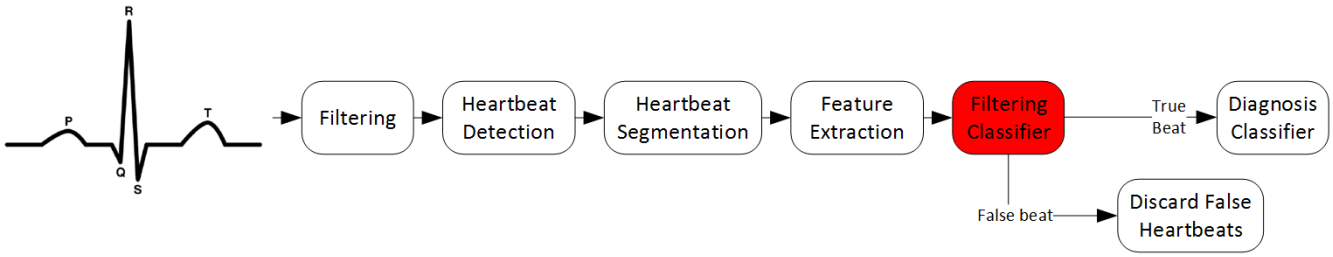


Figure 4.1: Extended ECG analysis flow

For the filtering classifier, we chose to use again an SVM based classifier, for the same reasons stated in the previous chapter.

In order to produce the SVM model we need to form different training and testing data sets, then the ones used for the model of the diagnosis classifier. Again, we use all 45 ECG records selected from the MIT-BIH database and using the WQRS detector we get a set of 104581 heartbeats. The selection of WQRS as detector is made based on the fact that it has a good accuracy on true beats detection, a low missed beats detection, and at the same time detects the most false beats, compared to the other two available detectors (Table 4.1). The later is desired, so that the data sets of true and false beats are as less imbalanced as possible. For each heartbeat we need a vector of attributes and a target value. The vector of attributes is the feature vector which contains the DWT coefficients of the heartbeat and

is formed at the feature extraction stage. As target value, we label each heartbeat as either True or False. This label is given based on the Derived R matching algorithm described in the previous chapter. Applying this algorithm to the detected heartbeats, we get a data set of 100231 true heartbeats and 4350 false heartbeats. In order for the two sets to have comparable analogy of true and false beats, half of the true heartbeats and half of the false heartbeats are used to form the training data set, and the rest of the heartbeats are used to form the testing data set.

The same design space exploration procedure, described in the previous chapter, is applied for the selection of the feature vector which gives the SVM model with the best performance results. The performance metrics are in this case adjusted as follows:

- Sensitivity = $\frac{\text{Num. of correctly classified Positive points}}{\text{Total number of Positive points}}$

  to be maximized i.e. the best possible recognition of True heartbeats is achieved.

- Specificity = $\frac{\text{Num. of correctly classified Negative points}}{\text{Total number of Negative points}}$

  to be maximized i.e. the best possible recognition of False heartbeats is achieved.

- Accuracy = $\frac{\text{Number of correctly classified points}}{\text{Total number of points}}$

  to be maximized

- Minimizing the computational overhead inflicted by the extra SVM false beat classifier on the overall ECG analysis flow to sustain real-time operation.

The main challenge of the target classification problem is that false beats are much less in number compared to true beats. This creates a very imbalanced training data set and if no action is taken to take this into account during training phase then the classifier fails at acceptably classifying false beats. In other words, since the goal of the training phase is to adequately train the classifier according to the input dataset, if a classifier which can successfully identifies only true beats is produced, then the overall classification accuracy of the model is high since true beats vastly dominate the training data set.

In cases of imbalanced datasets, accuracy is not the most reliable metric, since it can be misleading in the way described above. So we rather focus on the other three metrics,

sensitivity, specificity and computational cost.

Fig. 4.2 presents the results of the design space exploration applied on all the different combinations of the 8 sets of DWT coefficients. We can see that in all cases sensitivity is above 99.75% or in other words the True beats misclassified as False are relatively low. Furthermore, specificity i.e. the False beats successfully filtered is above 72% and reaching up to 88%.

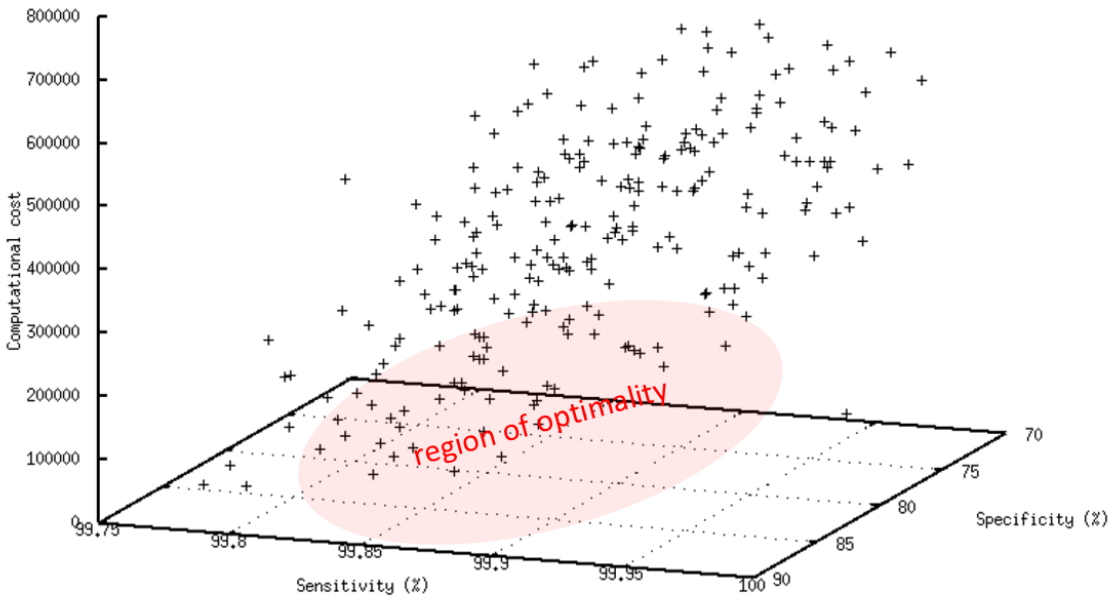


Figure 4.2: Design space exploration of pre-diagnosis filtering classifier

In an effort to determine the best of the design alternatives, we highlighted the region of Pareto optimal design points which maximize sensitivity and specificity while minimizing computational requirements of the trained classifier. The most medically reasonable choice is to decide the design alternative which maximizes sensitivity in order to reduce the True beats classified as False as much as possible. Inevitably, this impacts on the classifier's ability to filter false beats. Consequently if ones takes into account the tuple (sensitivity, specificity, computations), this results in (99.992%, 65.56%, 272900). However, we notice that a small reduction in sensitivity leads to a design alternative with (99.88%, 85.1%, 94512) which means that an increase 19.54% in specificity was achieved by a classifier which is 3 times less computationally demanding by sacrificing only 0.112% in sensitivity. In absolute numbers, this configuration has misclassified only 60 True beats out of testing data set of

50116. For even smaller computational cost, the tuple (99.85%, 82.62%, 30744) forms yet another design alternative.

4.3 Results Evaluation

Table 4.2: Baseline vs extended diagnosis flow

Analysis flow	Accuracy (%)	Sensitivity (%)	Specificity (%)	Number of False alarms
Baseline	97.87	99.40	99.40	2175
Extended	97.86	99.35	99.46	550

Table 4.2 summarizes comparative results between the typical and proposed ECG analysis flows. The feature vector used on the diagnosis classification stage of both flows is the one selected from the DSE of the diagnosis filtering classifier. The feature vector of the filtering classifier consists of the detail coefficients of the 3rd and 4rth level of decomposition (cD3, cD4), and corresponds to the tuple (99.97%, 69.98%, 92560).

Taking a closer inspection on the results, we can see that accuracy of the extended diagnosis flow is 0.01 less compared to the baseline diagnosis flow. To explain that, we will focus on the other metrics. Regarding sensitivity, the extended diagnosis flow lacks 0.05% compared to the baseline one. This is because, there is a small number of True beats who are erroneously filtered as False and therefore sensitivity of the complete flow decreases. On the contrary, due to the fact that the filtering classifier discards a large number of False beats, the specificity of the extended diagnosis flow is increased compared to the baseline one. We do not observe a steep rise due to the fact that most of the discarded False beats are successfully classified as Abnormal in the baseline flow. Apart from the provided metrics regarding the behavior of the two diagnosis flows, the greatest advantage of the filtering classifier is that, supposing that the diagnosis flow raises an alarm whenever a heart beat is considered to be Abnormal, there is about 75% reduction in these alarms owned to False beats, i.e. False alarms. The trade-off is that the filtering classifier imposes an average 27% in the required execution time of the complete diagnosis flow.

Chapter 5

ECG Analysis Flow on Embedded IoT Platform

5.1 IoT Platform

The Internet of Things (IoT) is increasingly being recognized by researchers and analysts as one of the most sophisticated technologies that has the potential to not only affect the health, safety and productivity of billions of people but also has a major economic impact. It primarily consists of physical objects or "things"that are embedded with sensors, actuators, computing devices and data communication capabilities. These are linked to networks for data transportation. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. Multiple traditional fields of technologies such as embedded systems, wireless sensor networks, control systems and others, all contribute to enabling the Internet of Things.

Healthcare is one of the most rapidly expanding application areas of IoT technology. Remote health management, managing lifestyle-related diseases and conditions, fitness programs, care at home, chronic diseases and care for the elderly are some of the important use cases. IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers or advanced hearing aids. Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens and of chronic patients, while also ensuring that proper treatment is being administered.

Fig. 5.1 demonstrates an IoT-based ECG monitoring design. Electrodes are used to sense the ECG voltage on the skin, amplifiers to increase the magnitude of the signal, and ADC to get digital samples. The signal is then sent to a wearable IoT based device to be processed. The data transfer can be conducted via a Bluetooth module. The output of the computing devise, the heartbeat diagnosis in the case of ECG signal classification, is finally sent via Internet connection to a remote device for further analysis and storage.

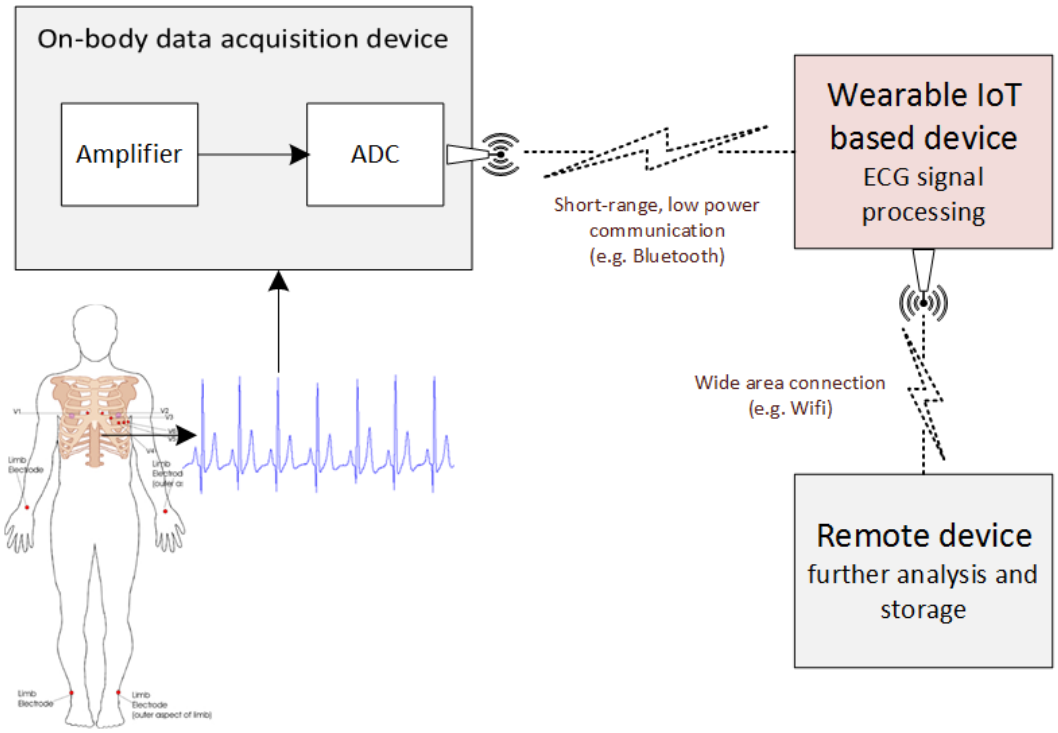


Figure 5.1: Iot-based ECG monitoring design

For the purposes of the current study, we choose to implement the ECG signal analysis and classification algorithm developed, on the Intel Galileo board. The Galileo is the first product to feature the Intel Quark SoC X1000, a chip designed for small-core products and low power consumption, and targeted at markets including the Internet of Things and wearable computing. The Quark SoC X1000 is a 32-bit, single core, single-thread, Pentium (P54C/i586) instruction set architecture (ISA)-compatible CPU, operating at speeds up to 400 MHz. The use of the Pentium architecture gives the Galileo the ability to run a fully-fledged Linux kernel. What's more, an on-board Ethernet port provides network connectivity, while also the underside provides a mini-PCI Express slot, designed for use with Intel's wireless network cards to add cheap Wi-Fi connectivity to designs.

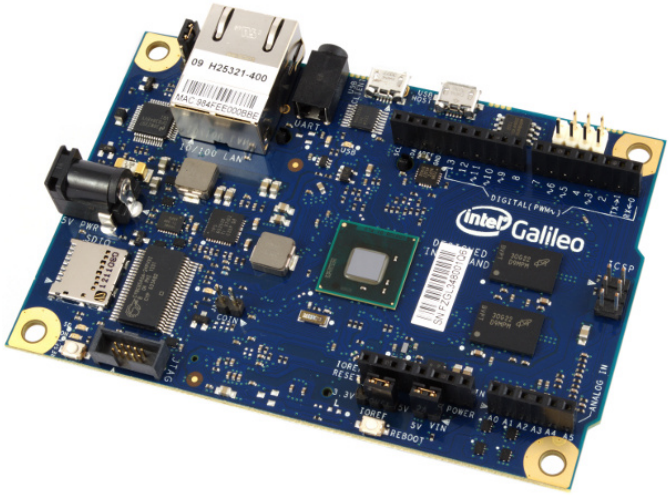


Figure 5.2: Intel Galileo board [5]

The Galileo board's technical specifications:

- **Operating system:** Yocto Project-based Linux
- **Processor:** Single-Core 400MHz Intel Quark X1000
- **Memory:** 256MB RAM
- **Dimensions:** 107mm x 74mm x 23mm
- **Weight:** 50g (excluding PSU)
- **GPIO:** 14x Digital Input/Output Pins, 6x Analogue Input Pins
- **Networking:** 1x Wired 10/100 Ethernet, Optional PCIe Wireless
- **Expansion:** USB 2.0 Host, Micro-SD Card

The Quark X1000 features:

- Up to 400MHz clock speed
- 16KB L1 Cache
- 512KB SRAM
- Single core, single thread

- Integrated SDIO, UART, SPI, USB, I2C, Ethernet, RTC

5.2 Implementation in C

Since in the first part of this study, the algorithmic analysis was done in Matlab environment, the code had to be converted in C, in order to continue with its implementation on the IoT platform. The online ECG signal processing and classification is the real-time part of the analysis that will be implemented on the embedded platform, while the offline training procedure (see fig. 3.2) is considered to already have been implemented on some other platform. The different stages of the analysis flow are implemented in C, as described below, and combined into a single program. The program reads sample by sample a digitized (at 360 samples per second) ECG signal. The analysis flow is executed for every set of 3000 samples that is read.

1. Filtering

   The signal, consisting of 3000 samples, is firstly filtered. For the filtering stage, the same FIR filters that were used in the Matlab environment are implemented in C.

2. Heartbeat detection

   Following, the stage of heartbeat detection is implemented. In the Matlab implementation we use two functions provided by PhysioNet in the WFDB Toolbox for Matlab, *wqrs*() and *ecgpuwave*(). PhysioNet also provides the source codes of these two functions, wqrs in C and ecgpuwave in Fortran. Since we only have wqrs in C, we alter the algorithm to only apply wqrs to the signal, with no substantial divergence in the output of this stage. Wqrs is a QRS complex detection function, that returns the approximate position of the QRS complex onset. Instead of passing this information to ecgpuwave, in order to get the exact position of the R peak and then apply the 257-sample window on the next stage of the heartbeat segmentation, we keep the QRS onset information as the heartbeat reference point, and adapt the window to cover the heartbeat waveform.

Wqrs analyzes an ECG signal, detecting QRS onsets, using a nonlinearly-scaled ECG curve length feature. The algorithm averages the first 8 seconds of the length-transformed input to determine on some initial thresholds that it uses. This is the reason we choose a 3000-samples (8.33 seconds) waveform as the minimum input of our program.

3. Heartbeat segmentation

   For the implementation of the heartbeat detection stage, we adapt the window to cover the PR and the QT intervals (see fig. 2.3). Based on statistics over the ECG waveform [35], we decide on a window of 86 samples before the QRS onset, and 170 samples after the QRS onset (window width of 257 samples). Comparing the output of the two implementations up to this stage, we see that the resulted heartbeat is satisfactory in both the Matlab and the C implementation, as demonstrated in fig. 5.3 for a random ECG waveform. The heartbeats detected in the 3000-sample signal are passed one by one to the next stages.

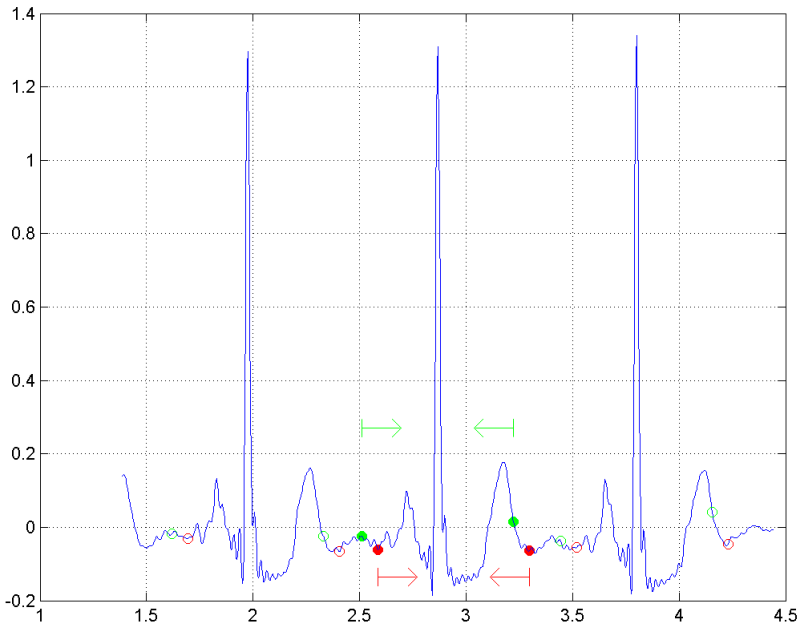


Figure 5.3: Heartbeat segmentation output for random ECG waveform: the green circles show the begining and end of the heartbeat as determined by the Matlab implementation, and the reb circles show the begining and end of the heartbeat as determined by the implementation in C.

4. Feature extraction

For the stage of feature extraction the Matlab function *wavedec*() is implemented in C. We use Matlab function $wfilters()$ to acquire the wavelet decomposition low-pass and high-pass filters associated with wavelet Daubechies of order 2 ('db2'). We apply symmetric-padding (boundary value symmetric replication) to the signal, since it is the default discrete wavelet transform extension mode of Matlab. Then, the convolution of the signal with each filter is implemented to produce the approximation and detail coefficients for each of 4 levels of decomposition, according to the following process: Given a signal s of length n, the DWT consists of $\log_2$ n stages at most. The first step produces, starting from s, two sets of coefficients: approximation coefficients CA1, and detail coefficients CD1. These vectors are obtained by convolving s with the low-pass filter Lo_D for approximation, and with the high-pass filter Hi_D for detail, followed by dyadic decimation (downsampling). The length N of each filter is equal to the order of the Daubechies wavelet. The signals F and G are of length $n + 2N - 1$ and the coefficients cA1 and cD1 are of length $floor((n - 1)/2) + N$. The next step splits the approximation coefficients cA1 in two parts using the same scheme, replacing s by cA1, and producing cA2 and cD2, and so on (fig. 5.4).

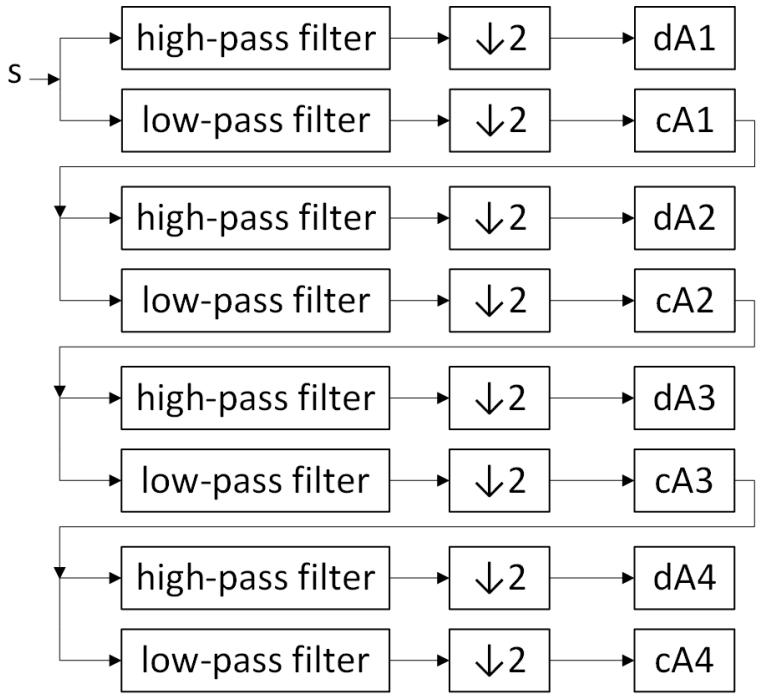


Figure 5.4: 4 levels of DWT

This stage is implemented parametrically in terms of the coefficients that compose the feature vector of the heartbeat used in the classification stage. For each decomposition level, the coefficients produced are stored in a matrix if they are part of the final feature vector, or else they are only used for the computation of the next decomposition level. The process stops whenever all required coefficients have been produced.

5. Classification

   For the final stage of classification, we use the LIBSVM library which includes the source code of *svmpredict*() in C. We convert the SVM model under examination produced by the same function in the Matlab environment into the format that is used in the C implementation, using a generator. In the beginning of the program, the function *load_svm_model*(), which is included in the LIBSVM library, is called to load the SVM model from that file that we have created. The SVM model is only created once, in that initialization section of the program, and used in the classification stage for each heartbeat.

In fig. 5.5 we illustrate the overall framework. This includes the exploration phase conducted in the Matlab environment, meaning the DSE which resulted in a set of Pareto solutions. In the customization phase, having selected a desirable solution, the feature vector is set accordingly and the corresponding SVM model is produced by a generator which converts the Matlab SVM model to the appropriate format for the C implementation. Finally, the runtime phase is the implementation of the algorithm in C, for the configuration selected, conducted on the Galileo board.

5.3 Experimental Results

From the offline training procedure, we get the SVM model which is being used at the classification stage of the analysis flow (see fig. 3.2). The configurations of this model, derived from the design space exploration for the SVM tuning, described in chapter 3.3. The metrics taken into consideration for the selection were, as mentioned there, accuracy

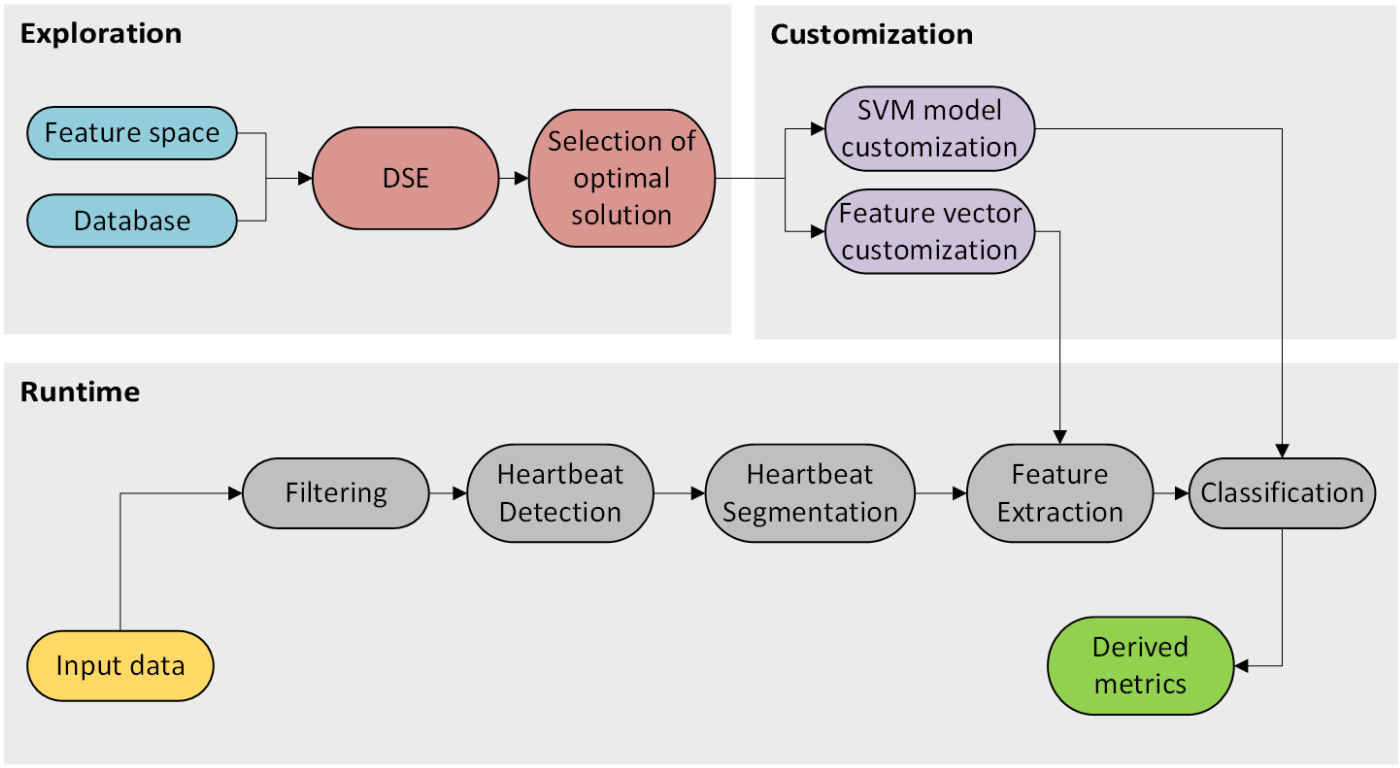


Figure 5.5: Structure of algorithm implemented in C.

and computational cost. The first, is independent of the implementation platform, while the latter depends on the system resources. During the DSE, the computational cost was not measured as the absolute execution time of heartbeat classification on the platform used, but it was rather computed theoretically as the product of number of support vectors and size of feature vector, since it is the number of multiplications required for a new heart beat to be classified, as mentioned already. In order to verify these theoretical computations with the absolute execution time of heartbeat classification on the embedded platform, and make the final SVM model selection, we selected the 10 best , the 10 worst, and 11 configurations from inbetween (Table 5.1), as resulted from the DSE, to implement on the platform.

The results of the time averages and the percentage of total execution time, for each stage of the analysis flow, for the selected configurations, are shown in Table 5.2. The time averages for the filtering and the heartbeat detection stage, correspond to the 3000 sample input, while the time averages of the feature extraction and the classification stage are per heartbeat detected. As input we used 101*3000 samples (303000 out of 650000 samples contained in each ECG signal record) from 43 records (out of 48 total records of MIT-BIH

database). In the total of 4343 sets of 3000 samples inputed to each configuration, 45529 heartbeats were detected each time, and it was seen that approximately 10 heartbeats were contained in each set of 3000 samples. Therefore, for the estimation of the percentage of the total execution time of each stage, it was assumed that for every 3000 sample input, 10 heartbeats are detected. In the best cases examined, the total execution time is much less than the approximate 9 seconds required for the 3000 samples of the signal to be read. This means that the ECG signal analysis and classification proposed, can be performed in real-time.

As we can see in fig. 5.6, the scaling of execution time of the classification stage for the different configurations, is as expected from the theoretical approach, meaning the computations required in each configuration.

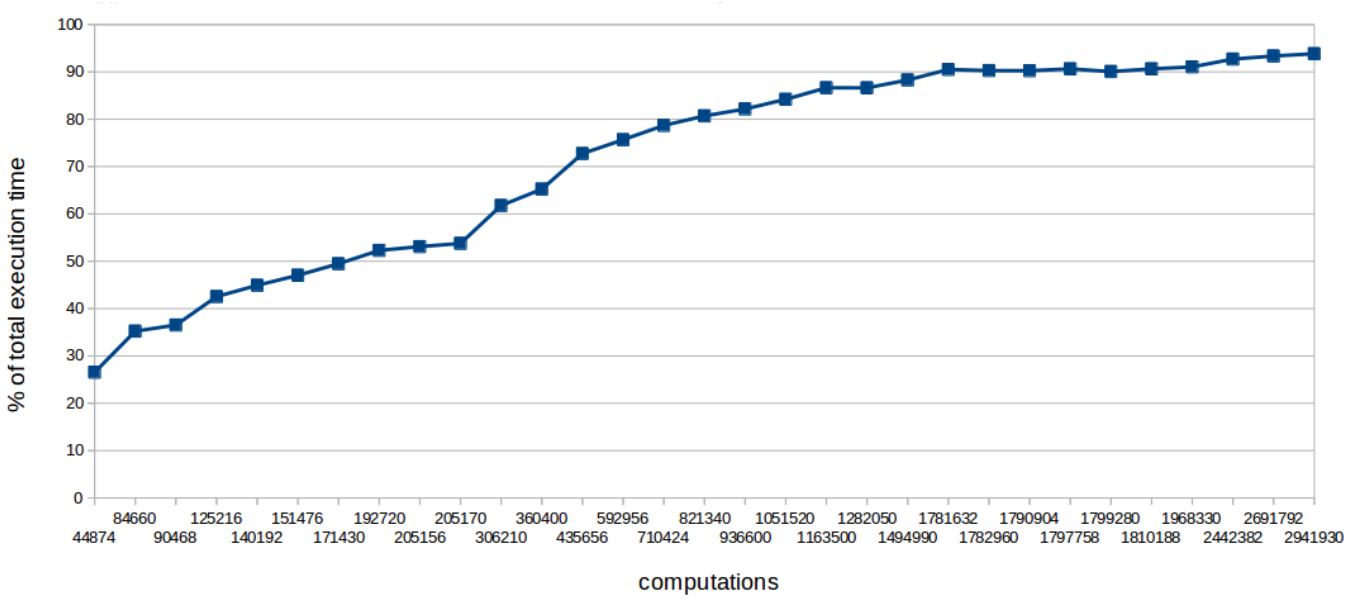


Figure 5.6: Scaling of execution time in accordance with the computations required in each configuration.

In fig. 5.7, we present the average % of total execution time for each stage of the algorithm, of the 10 best configurations examined. Fig. 5.8 presents the corresponding % execution times of each stage, for 11 configurations from the inbetween. Fig. 5.9 presents the same % execution times, for the 10 most computationally demanding configurations. The classification stage takes up more than 90% of the total execution time.

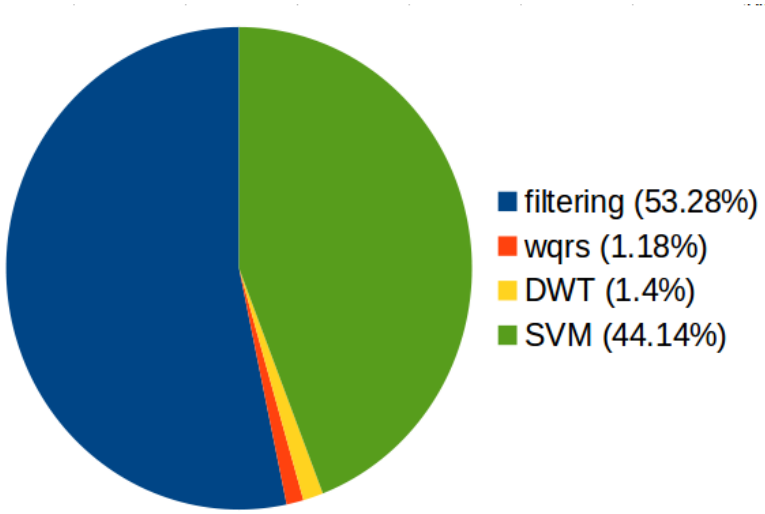


Figure 5.7: Avarege % of total execution time for each stage, for the 10 best configurations.

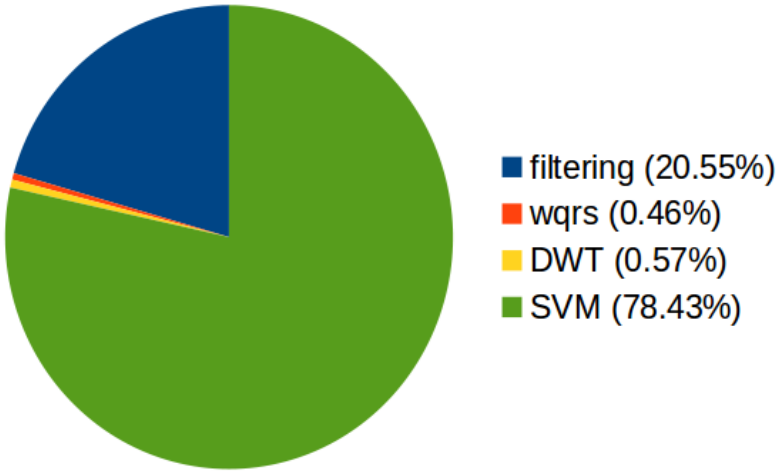


Figure 5.8: Average % of total execution time for each stage, for 11 configurations from inbetween.

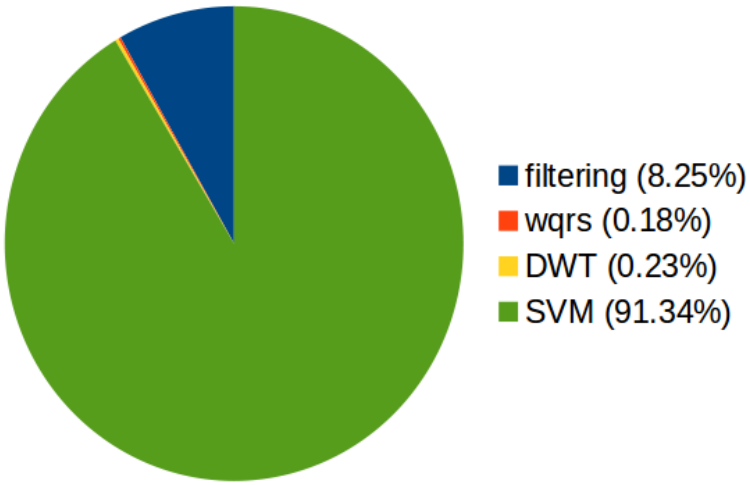


Figure 5.9: Average % of total execution time for each stage, for the 10 most computationally demanding configurations.

Table 5.1: Selected configurations from DSE

configuration	DWT coefficients	number of support vectors	feature vector size	Accuracy (%)	Computational cost
1	cA4	2493	18	98.99	44874
2	cA3	2490	34	98.93	84660
3	cA4, cD4	2513	36	98.90	90468
4	cA3, cA4	2408	52	98.99	125216
5	cA3, cD4	2696	52	98.80	140192
6	cA4, cD3	2913	52	98.62	151476
7	cA3, cA4, cD4	2449	70	98.88	171430
8	cA2	2920	66	98.62	192720
9	cA3, cD3	3017	68	98.67	205156
10	cA4, cD3, cD4	2931	70	98.62	205170
11	cA2, cA3, cA4	2595	118	98.84	306210
12	cA4, cA3, cA4, cD4	2650	136	98.79	360400
13	cD3, cD4	8378	52	94.83	435656
14	cA3, cA4, cD1	3258	182	98.36	592956
15	cA4, cD2, cD3, cD4	3861	184	98.00	710424
16	cA3, cA4, cD1, cD3, cD4	3510	234	98.26	821340
17	cA1, cA2, cA3, cA4, cD3, cD4	3122	300	98.60	936600
18	cA2, cD1, cD3, cD4	4240	248	97.70	1051520
19	cD2, cD3	11635	100	93.25	1163500
20	cA2, cA3, cD1, cD2, cD3	3885	330	98.02	1282050
21	cA1, cA3, cD1, cD2, cD4	3955	378	97.81	1494990
22	cA1, cA2, cA3, cA4, cD1, cD2, cD3, cD4	3592	496	98.22	1781632
23	cA1, cA2, cA3, cD1, cD2, cD3	3876	460	98.13	1782960
24	cA1, cA2, cD1, cD2, cD3	4204	426	97.78	1790904
25	cA1, cA2, cA3, cD1, cD2, cD3, cD4	3761	478	98.01	1797758
26	cA1, cD1, cD2, cD3, cD4	4760	378	97.28	1799280
27	cA1, cA2, cD1, cD2, cD3, cD4	4077	444	97.65	1810188
28	cD1, cD3, cD4	10815	182	92.70	1968330
29	cD1, cD2, cD4	11413	214	92.04	2442382
30	cD1, cD2, cD3, cD4	10854	248	92.45	2691792
31	cD1, cD2, cD3	12791	230	90.64	2941930

Table 5.2: Results from the implementations on Intel Galileo

	filtering		beat detection		feature extraction		classification	
configuration	average (in sec)	% of total ex. time	average (in sec)	% of total ex. time	average (in sec)	% of total ex. time	average (in sec)	% of total ex. time
1	0.77127	70.01	0.01691	1.53	0.00210	1.91	0.02925	26.55
2	0.77807	61.88	0.01707	1.36	0.00192	1.53	0.04430	35.23
3	0.62330	60.49	0.01373	1.33	0.00171	1.66	0.03762	36.52
4	0.62322	54.80	0.01381	1.21	0.00166	1.46	0.04836	42.53
5	0.62386	52.48	0.01379	1.16	0.00171	1.44	0.05341	44.92
6	0.62326	50.45	0.01380	1.12	0.00173	1.40	0.05811	47.03
7	0.65056	48.14	0.01451	1.07	0.00178	1.32	0.06685	49.47
8	0.68187	45.72	0.01504	1.01	0.00147	0.99	0.07799	52.29
9	0.62327	44.76	0.01396	1.00	0.00158	1.13	0.07394	53.10
10	0.62284	44.03	0.01385	0.98	0.00173	1.22	0.07604	53.76
11	0.62648	36.43	0.01390	0.81	0.00171	0.99	0.10622	61.77
12	0.62396	33.13	0.01388	0.74	0.00171	0.91	0.12284	65.23
13	0.62414	25.97	0.01386	0.58	0.00170	0.71	0.17479	72.74
14	0.71297	23.17	0.01594	0.52	0.00199	0.65	0.23284	75.67
15	0.62899	20.30	0.01380	0.45	0.00177	0.57	0.24380	78.68
16	0.62782	18.38	0.01395	0.41	0.00176	0.51	0.27556	80.69
17	0.65934	16.99	0.01470	0.38	0.00184	0.47	0.31877	82.15
18	0.62769	15.04	0.01393	0.33	0.00175	0.42	0.35130	84.20
19	0.73301	12.75	0.01622	0.28	0.00185	0.32	0.49811	86.65
20	0.65047	12.75	0.01433	0.28	0.00173	0.34	0.44180	86.63
21	0.71573	11.15	0.01593	0.25	0.00206	0.32	0.56645	88.28
22	0.62523	09.02	0.01401	0.20	0.00181	0.26	0.62759	90.52
23	0.62374	09.28	0.01376	0.20	0.00169	0.25	0.60676	90.27
24	0.62314	09.29	0.01374	0.20	0.00171	0.25	0.60542	90.25
25	0.69495	08.90	0.01530	0.20	0.00204	0.26	0.70780	90.64
26	0.62363	09.45	0.01389	0.21	0.00179	0.27	0.59465	90.07
27	0.63113	08.91	0.01399	0.20	0.00183	0.26	0.64231	90.64
28	0.62258	08.50	0.01389	0.19	0.00177	0.24	0.66672	91.06
29	0.62295	06.95	0.01380	0.15	0.00177	0.20	0.83134	92.70
30	0.70772	06.32	0.01552	0.14	0.00199	0.18	1.04530	93.36
31	0.65894	05.89	0.01472	0.13	0.00176	0.16	1.05047	93.83

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this work, we develop an algorithm for ECG analysis and classification, and implement it on an IoT-based embedded platform. This algorithm is our proposal for a wearable ECG diagnosis device, suitable for 24-hour continuous monitoring of the patient.

The proposed algorithm is composed of five stages: filtering, heartbeat detection, heartbeat segmentation, feature extraction, classification. The Discrete Wavelet Transform was used for feature extraction, whereas a Support Vector Machine based classifier was adopted for the heartbeat classification. Design space exploration conducted on different feature vectors describing a heart beat, resulted in a variety of classifiers with different accuracy and computational characteristics. In almost all cases, the accuracy achieved is above 97%. The best result comes of the feature vector which contains the approximate coefficients of the 4th level of decomposition, with 98.9% accuracy, a feature vector of size 18, and 2493 support vectors.

Implementing the 10 best configurations of the design space exploration, on the Galileo board, we demonstrate that the computational cost is such, that the ECG analysis and classification can be performed in real-time.

Furthermore, we analyzed the ability of R peak detectors to successfully identify R peaks in an ECG signal. Based on a significant number of erroneously identified peaks we proposed a Support Vector Machine based classifier to be incorporated in the ECG analysis flow in order to identify and discard these false R peaks. The proposed false beat filtering classifier is able to filter up to 75% of false R peaks while keeping the number of True beats misclassified as False less than 0.01%. The required high accuracy creates a complex classifier but

design space exploration on different feature vectors enabled us to keep its computational requirements low, resulting in 27% in the execution time of the ECG analysis flow.

6.2 Future Work

Our work focuses on the development of a typical analysis flow over a one-lead ECG signal, distinguishing two heartbeat groups (Normal - Abnormal). This flow can be expanded by examining more heartbeat classes, as these are suggested by the MIT-BIH database, using multi-class SVMs. The implementation of the algorithmic flow in C code, can be completed by adding the extra stage of the filtering classifier proposed in the extended structure of the analysis.

Furthermore, the *ecgpuwave* function, provided in Fortran and in Matlab code, and used in the Matlab implementation, can be converted in C code, and be included in the C implementaion. This would be especially useful if a different method for feature extraction is used, such as waveform morphology, where more information on fiducial points is needed than that obtained by only using the $wqrs$ function.

Another major course of action, would be the energy analysis of the application implemented on the Galileo board. For the target platform being a wearable device, the computations involved in the application must be achieved at very low power levels (e.g. 1-10 mW). The satisfying execution times achieved in this study, allow the examination of different methods to reduce power consumption to a desirable level. One such method would be to underclock the CPU.

Finally, this work can be extended by completing the IoT-based ECG monitoring design shown in 5.1. This would involve the implementation of the input data transfer from the on body data acquisition device to the application through a Bluetooth module, as well as the implementation of the wide area communication between the application running on the wearable device and a remote server, where the output data would be stored.

References

[1] R. E. Klabunde, *Cardiovascular Physiology Concepts*. September 2011. xxixxi, 7

[2] D. Price, "How to read an electrocardiogram (ecg)," *South Sudan Medical Journal*, 2010. xxixxi, 9

[3] "Electrocardiography." `https://en.wikipedia.org/wiki/Electrocardiography`. xxixxi, 10

[4] V. Tsoutsouras, D. Azariadi, S. Xydis, and D. Soudris, "Effective learning and filtering of faulty heart-beats for advanced ecg arrhythmia detection using mit-bih database," December 2015. xxixxi, 24

[5] "Galileo getting started guide." `https://learn.sparkfun.com/tutorials/galileo-getting-started-guide`. xxixxi, 38

[6] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 3, pp. 45–50, 2001. 2

[7] P. de Chazal, M. O'Dwyer, and R. B.Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1196–1206, July 2004. 3, 29

[8] P. de Chazal and R. Reilly, "A patient-adapting heartbeat classifier using ecg morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2535–2543, 2006. 3

[9] E. D. Übeyli, "Ecg beats classification using multiclass support vector machines with error correcting output codes," *Digital Signal Processing*, vol. 17, no. 3, pp. 675–684, 2007. 3, 21, 29

[10] S.-N. Yu and Y.-H. Chen, "Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network," *Pattern Recognition Letters*, vol. 28, pp. 1142–1150, 2007. 3

[11] D. Benitez, P. Gaydecki, A. Zaidi, and A. Fitzpatrick, "The use of the hilbert transform in ecg signal analysis," *Comput. Biol. Med.*, vol. 31, pp. 399–406, 2001. 3

[12] K. Minami, H. Nakajima, and T. Toyoshima, "Real-time discrimination of ventricular tachyarrhythmia with fourier-transform neural network," *IEEE Trans. Biomed. Eng.*, vol. 46, pp. 179–185, 1999. 3

[13] M. Lagerholm, G. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo, "Clustering ecg complex using hermite functions and self-organizing maps," *IEEE Trans. Biomed. Eng.*, vol. 47, pp. 838–848, 2000. 3, 29

[14] P. Laguna, R. Jane, S. Olmos, N. Thakor, H. Rix, and P. Caminal, "Adaptive estimation of qrs complex wave features of ecg signal by the hermite model," *Med. Biol. Eng. Comput.*, vol. 34, pp. 58–68, 1996. 3

[15] A. Khazaee and A. Ebrahimzadeh, "Classification of electrocardiogram signals with support vector machines and genetic algorithms using power spectral features," *Biomedical Signal Processing and Control*, vol. 5, pp. 252–263, 2010. 3

[16] K. L, A.-F. AS, and B. S., "A quantitative analysis approach for cardiac arrhythmia classification using higher order spectral techniques," *IEEE Trans. Biomed. Eng.*, vol. 52, pp. 1840–1845, 2005. 3

[17] E. D. Übeyli, ""detecting variabilities of ecg signals by lyapunov exponents," *Neural computing and applications*, vol. 18, no. 7, pp. 653–662, 2009. 3

[18] W. K. Lei1, B. N. Li1, M. C. Dong, and M. I. Vai, "Afc-ecg: An adaptive fuzzy ecg classifier a.," *(Eds.): Soft Computing in Industrial Applications*, vol. ASC 39, pp. 189–199, 2007. 3

[19] U. R. Acharya, P. S. Bhat, S. S. Iyengar, A. Rao, and S. Dua, "Classification of heart rate data using artificial neural network and fuzzy equivalence relation," *Pattern Recognition*, vol. 36, pp. 61–68, 2003. 3

[20] I. Güler and E. D. Übeyli, "Ecg beat classifier designed by combined neural network model," *Pattern Recognition*, vol. 38, no. 2, pp. 199–208, 2005. 3, 29

[21] B. Anuradha and V. C. V. Reddy, "Ann f classification of cardiac arrhythmias," *ARPN Journal of Engineering and Applied Sciences*, vol. 3, no. 3, pp. 1–6, 2008. 3

[22] D. Coast, R. Stern, G. Cano, and S. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 9, pp. 826–836, 1990. 3

[23] L. Chengwei, W. Shoubin, X. Aijun, and P. Hui, "Clinical diagnosis of cardiac disease based on support vector machine," *IFMBE Proceedings*, vol. 14, no. 2, pp. 1273–1276, 2007. 3

[24] B. Mohammadzadeh, S. Kamaledin, Setarehdan, and M. Maryam, "Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal," *Artificial Intelligence in Medicine*, vol. 44, no. 1, pp. 51–64, 2008. 3, 29

[25] K. Polat, B. Akdemir, and S. Gune, "Computer aided diagnosis of ecg data on the least square support vector machine," *Digital Signal Processing*, vol. 18, pp. 25–32, 2008. 3

[26] Y.-C. Yeh, C. W. Chiou, and H.-J. Lin, "Analyzing ecg for cardiac arrhythmia using cluster analysis," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1000–1010, 2012. 3, 29

[27] "R. mark and g. moody, mit-bih arrhythmia database,." Available online from: `http://www.physionet.org/physiobank/database/mitdb/`. 12

[28] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit,

and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. 19

[29] D. K. Ojha and M. Subashin, "Analysis of electrocardiograph (ecg) signal for the detection of abnormalities using matlab," *World Academy of Science, Engineering and Technology International Journal of Medical, Health, Biomedical and Pharmaceutical Engineering*, vol. 8, no. 2, 2014. 20

[30] P. Laguna, R. Jané, and P. Caminal, "Automatic detection of wave boundaries in multilead ecg signals: Validation with the cse database," *Computers and Biomedical Research*, vol. 27, no. 1, pp. 45–60, 1994. 21

[31] R. Jané, A. Blasi, J. García, and P. Laguna, "Evaluation of an automatic threshold based detector of waveform limits in holter ecg with the qt database," *Computers in Cardiology*, vol. 24, pp. 295–298, 1997. 21

[32] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990. 21

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 20(3):273–297, 1995. 22, 27

[34] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011. 22

[35] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods And Tools for ECG Data Analysis*. January 2006. 40