

دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده‌ی مهندسی کامپیوتر و فناوری اطلاعات

پروژه‌ی کارشناسی مهندسی کامپیوتر

گرایش معماری کامپیوتر

طراحی و پیاده‌سازی یک سیستم تشخیص بی‌درنگ آریتمی
قلبی بر بستر اینترنت اشیا

نگارش
مرضیه تاجیک

استاد راهنما
دکتر محمود ممتازپور

استاد داور
دکتر مرتضی صاحب‌زمانی

اسفند ۱۳۹۷

چکیده

تشخیص آریتمی‌های قلبی در صورتی که به موقع انجام شود، می‌تواند از عواقب خطرناک بعدی این دسته از بیماری‌ها جلوگیری نماید. هدف از این پروژه، طراحی یک سیستم تشخیص آریتمی است که بتواند به صورت بی‌درنگ ضربان‌ها را دریافت کرده و آن‌ها را دسته‌بندی کند. در این پروژه بستری برای تشخیص بی‌درنگ آریتمی قلبی طراحی و پیاده‌سازی شده است. این سیستم با استفاده از اینترنت اشیا و در دو بخش طراحی می‌شود. بخش اول پیش‌پردازش‌هایی را بر روی سیگنال ضربان قلب دریافت‌شده از بیمار انجام می‌دهد و پیاده‌سازی آن بر روی یک میکروکنترلر است. در بخش دوم، پردازش‌های اصلی شامل استخراج ویژگی‌های سیگنال پیش‌پردازش‌شده و اجرای یک الگوریتم دسته‌بندی بر روی آن‌ها، بر روی یک سرور پیاده‌سازی می‌شود. این دو بخش به کمک اینترنت با یکدیگر ارتباط دارند. الگوریتم دسته‌بندی استفاده‌شده می‌تواند با دقت میانگین ۰/۴۴، حساسیت ۰/۷۳ و اندیس J_k برابر با ۰/۴۳، کلاس آریتمی ضربان‌های ارسال شده را تشخیص دهد. زمان پاسخ سیستم از زمان دریافت نمونه‌ی سیگنال نوار قلب تا تشخیص آریتمی حدود ۲ ثانیه تخمین زده می‌شود، همچنین در این کار یک رابط کاربری در قالب اپلیکیشن وب طراحی و ساخته شده است که نتایج تشخیص آریتمی آخرین ضربان‌های دریافت‌شده را به پزشک و بیمار نمایش می‌دهد.

واژگان کلیدی: اینترنت اشیا، آریتمی قلبی، الگوریتم دسته‌بندی، تشخیص بی‌درنگ

فهرست مطالب

فصل ۱	مقدمه.....	۱
۱-۱	ضرورت تشخیص خودکار آریتمی قلبی.....	۲
۲-۱	تعریف صورت مسئله.....	۲
۳-۱	نیازمندی‌های پروژه.....	۲
۴-۱	راه حل ارائه شده.....	۳
فصل ۲	مفاهیم اولیه.....	۵
۱-۲	قلب و نحوه‌ی عملکرد آن.....	۶
۱-۱-۲	سیستم هدایت الکتریکی قلب.....	۶
۲-۲	آریتمی قلبی.....	۸
۱-۲-۲	انواع آریتمی قلبی.....	۸
۳-۲	سیگنال نوار قلب.....	۹
۱-۳-۲	نحوه‌ی قرارگیری الکترودها بر روی پوست و لیدهای تولیدشده.....	۹
۲-۳-۲	ترکیب QRS.....	۱۰
۴-۲	مسائل دسته‌بندی.....	۱۲
۱-۴-۲	روش ماشین بردار پشتیبانی (SVM).....	۱۳
۲-۴-۲	دسته‌بندی داده‌ها با استفاده از روش SVM.....	۱۷

۱۸.....	۵-۲ معیارهای کارایی در مسائل دسته‌بندی
۱۸.....	۱-۵-۲ ماتریس درهم‌ریختگی
۱۹.....	۲-۵-۲ دقت و حساسیت
۲۰.....	۳-۵-۲ صحت و صحت کلی
۲۱.....	فصل ۳ روش حل مسئله
۲۲.....	۱-۳ مقدمه
۲۲.....	۲-۳ عملیات پیش‌پردازش بر روی سخت‌افزار
۲۳.....	۱-۲-۳ مراحل تشخیص QRS
۲۷.....	۲-۲-۳ پیاده‌سازی الگوریتم تشخیص QRS بر روی بستر سخت‌افزاری
۲۹.....	۳-۳ عملیات پردازش سمت سرور
۲۹.....	۱-۳-۳ نحوه‌ی اجرای الگوریتم یادگیری
۳۲.....	۲-۳-۳ نحوه‌ی پردازش داده‌های دریافت‌شده در سرور
۳۳.....	۳-۳-۳ داده‌های مورد بررسی در الگوریتم یادگیری
۳۶.....	۴-۳-۳ ارزیابی نتایج حاصل از یادگیری
۳۹.....	فصل ۴ نتایج به‌دست‌آمده
۴۰.....	۱-۴ زمان پاسخ سیستم
۴۱.....	۲-۴ معیارهای کارایی نهایی الگوریتم دسته‌بندی
۴۲.....	فصل ۵ نتیجه‌گیری و کارهای آینده
۴۵.....	مراجع

فهرست شکل‌ها

۱-۱	مراحل اصلی یک سیستم خودکار تشخیص آریتمی [۱]..... ۴
۱-۲	سیستم هدایت الکتریکی قلب [۲]..... ۷
۲-۲	نوار قلب ۱۲ لیدی گرفته‌شده از یک فرد سالم [۳]..... ۱۰
۳-۲	ترکیب QRS [۴]..... ۱۱
۴-۲	نموداری از حل یک مسئله‌ی دسته‌بندی دوبعدی با روش SVM [۵]..... ۱۴
۵-۲	سمت چپ: داده‌های غیر قابل جداسازی توسط یک ابرصفحه در یک فضای دوبعدی، سمت راست: داده‌های انتقال‌داده‌شده به فضای سه‌بعدی و قابل جداسازی [۶]..... ۱۵
۶-۲	ماتریس درهم‌ریختگی برای یک مسئله‌ی دسته‌بندی چنددسته‌ای [۷]..... ۱۹
۱-۳	بخش پیش‌پردازش بر روی سخت‌افزار در مراحل تشخیص آریتمی..... ۲۲
۲-۳	بخش پردازش سمت سرور در مراحل تشخیص آریتمی..... ۲۹
۳-۳	خروجی نشان‌داده‌شده در مرورگر..... ۳۳

فهرست جدول‌ها

نتایج دسته‌بندی در کلاس‌های مختلف ضربان و به صورت میانگین.....	۴۱
۱-۴	

فصل ۱

مقدمه

۱-۱ ضرورت تشخیص خودکار آریتمی قلبی

بر اساس آمارهای سازمان سلامت جهانی^۱ بیماری‌های قلبی-عروقی^۲ رتبه‌ی اول را در بین بیماری‌های کشنده در سطح جهان دارند. برای مثال در سال ۲۰۱۶ حدود ۱۷/۹ میلیون مرگ (حدود ۳۱٪ آمار کلی فوت) به علت بیماری‌های قلبی عروقی تخمین زده شده‌است. [۸] حدود ۲۵٪ این تعداد را مرگ‌های ناگهانی قلبی^۳ تشکیل می‌دهند. [۹] در چنین شرایطی، بیمار در طول مدت یک ساعت پس از آغاز علائم دچار ایست قلبی می‌شود. علت اصلی ایست‌های قلبی ناگهانی، آریتمی‌های قلبی هستند. [۱۰] این عبارت به دسته‌ای از بیماری‌های قلبی اطلاق می‌شود که در آن‌ها، اختلالاتی در آهنگ طبیعی تپش قلب به وجود می‌آید. با وجود این که بیشتر آریتمی‌ها بی خطر هستند، در برخی موارد در صورت عدم رسیدگی می‌توانند مرگبار باشند. به همین دلیل، تشخیص و درمان به موقع آن‌ها از اهمیت بالایی برخوردار است.

۲-۱ تعریف صورت مسئله

در این پروژه هدف بر این است که بستری بی‌درنگ برای تشخیص انواع آریتمی‌های قلبی پیاده‌سازی شود. این بستر قابلیت این را دارد که با اتصال یک حس‌گر دیجیتال ضربان قلب به آن، ضربان‌های قلب را دریافت کرده، وجود یا عدم وجود آریتمی در هر ضربان، و همچنین در صورت وجود آریتمی، نوع آن را تشخیص داده و نتیجه‌ی این تشخیص را به اطلاع بیمار و پزشک او برساند. نیازمندی‌های چنین سیستمی در بخش بعدی به تفصیل شرح داده خواهند شد.

۳-۱ نیازمندی‌های پروژه

مهم‌ترین نیازمندی این بستر، بی‌درنگ بودن آن است. از چنین سیستمی انتظار می‌رود با دریافت هر ضربان قلب، در طول مدت‌زمان معینی عادی یا غیر عادی بودن آن را تشخیص دهد. مدت‌زمان بین لحظه‌ی ورود سیگنال ضربان قلب تا لحظه‌ای که نوع آن تشخیص داده شده و به اطلاع کاربر می‌رسد، محاسبه شده و بالاترین حد آن ضمانت می‌شود. چنین

^۱World Health Organization

^۲Cardiovascular diseases

^۳Sudden Cardiac Deaths (SCDs)

سیستمی از نوع بی‌درنگ نرم^۴ است، چرا که در صورت عدم تشخیص تعدادی ضربان پیش از ضرب‌الاجل^۵ تعیین‌شده، سیستم هم‌چنان می‌تواند به کار خود ادامه داده و ضربان‌های بعدی را پردازش کند.

به دلیل اهمیت تشخیص و درمان سریع در برخی از انواع خطرناک آریتمی، به خصوص آریتمی‌هایی که منجر به ایست ناگهانی قلبی می‌شوند، لازم است مراحل پردازش و ابزارهای مورد استفاده به نحوی بهینه‌سازی شوند که کمترین زمان ممکن برای تشخیص آریتمی و نوع آن در یک ضربان صرف شود. به این شکل در صورت وجود آریتمی در یک ضربان، پزشک معالج می‌تواند به سرعت از آن مطلع شده و اقدامات لازم را انجام دهد.

قابل حمل بودن و قابل استفاده‌ی آسان بودن توسط بیمار نیز، نیازمندی‌های دیگر این بستر هستند. در چنین کاربردی انتظار می‌رود بیمار دستگاهی ساده و قابل حمل^۶ در اختیار داشته‌باشد. در این شرایط، کم‌مصرف بودن دستگاه نیز اهمیت پیدا می‌کند، چرا که برای تامین انرژی از باتری استفاده خواهد شد و لازم است مصرف انرژی دستگاه طوری مدیریت شود که به شارژهای مکرر نیاز پیدا نکند.

این دستگاه ضربان قلب بیمار را دریافت کرده و نتیجه‌ی پردازش آن را از طریق یک اپلیکیشن موبایل یا وب به اطلاع خود بیمار یا پزشک او می‌رساند. اپلیکیشن طراحی‌شده برای نمایش نتایج به بیمار و پزشک، باید رابط کاربری مناسبی داشته‌باشد و تجربه‌ی کاربری خوبی را برای کاربر فراهم کند.

۴-۱ راه‌حل ارائه‌شده

در چند دهه‌ی گذشته پژوهش‌های گسترده‌ای بر روی طراحی سیستم‌های خودکار تشخیص آریتمی صورت گرفته‌است. در این سیستم‌ها، ابتدا سیگنال نوار قلب^۷ به وسیله‌ی الکترودها و تجهیزات مخصوص، از بیمار گرفته شده و فیلترهایی به جهت حذف انواع نویزها بر روی آن اعمال می‌شود. سپس یک الگوریتم قطعه‌بندی^۸ با هدف استخراج تک‌تک ضربان‌ها بر روی نوار قلب اجرا می‌شود. در مرحله‌ی بعد، مجموعه‌ای از ویژگی‌ها از هر یک از ضربان‌ها استخراج شده و به یک دسته‌بند^۹ داده می‌شود. این دسته‌بند نوع ضربان که خروجی نهایی این سیستم است را تعیین می‌کند. در این پروژه نیز معماری کلی

^۴Soft real-time

^۵Deadline

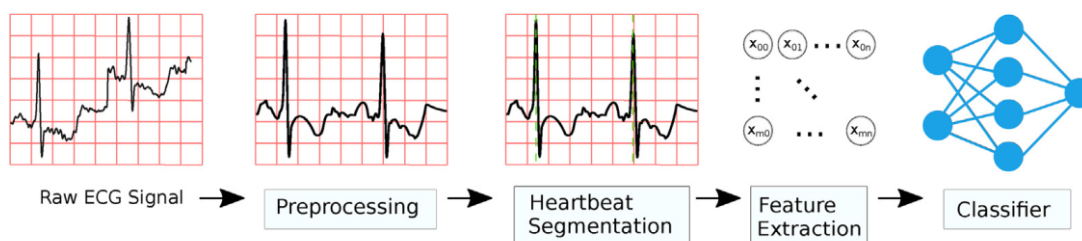
^۶Portable

^۷Electrocardiogram (ECG)

^۸Segmentation

^۹Classifier

سیستم پیاده‌سازی شده به همین نحو است. مراحل کلی تشخیص خودکار آریتمی در شکل ۱-۱ قابل مشاهده است.



شکل ۱-۱ مراحل اصلی یک سیستم خودکار تشخیص آریتمی [۱]

این روش در اکثر کارهای گذشته در زمینه‌ی تشخیص خودکار آریتمی به کار برده شده‌است. روش‌های متنوعی برای پیاده‌سازی الگوریتم دسته‌بندی استفاده شده‌اند، برای مثال در [۱۱] از سیستم‌های فازی^{۱۰}، در [۱۲] و [۱۳] از شبکه‌های عصبی مصنوعی^{۱۱} و در [۱۴] و [۱۵] از ماشین‌های بردار پشتیبانی^{۱۲} برای دسته‌بندی انواع آریتمی استفاده شده‌است. در این پروژه نیز از همین معماری کلی استفاده کرده‌ایم، با این تفاوت که سعی شده‌است راه‌حلی برای بی‌درنگ ساختن این سیستم ارائه شود.

همان‌طور که در بخش نیازمندی‌ها اشاره شد، بی‌درنگ بودن و قابل حمل بودن از نیازمندی‌های سیستم هستند. برای پاسخ‌گویی به این نیازمندی‌ها، اینترنت اشیا به عنوان راه‌حلی مناسب تشخیص داده می‌شود، چرا که به کمک آن می‌توان بخشی از سیستم را بر روی دستگاهی ساده و قابل حمل، بدون نیاز به توان پردازشی بالا پیاده‌سازی کرده و پردازش‌های سنگین‌تر را به عهده‌ی یک سرور قرار داد. اینترنت اشیا ارتباط بین این بخش‌ها را ممکن می‌سازد.

سیستم بی‌درنگ پیاده‌سازی شده در این کار، قادر به تشخیص نوع آریتمی هر یک از ضربان‌های گرفته‌شده از بیمار با دقتی بالا، و رساندن نتایج به کاربر است. در ادامه ابتدا به مفاهیم مورد استفاده در کار پرداخته و سپس روش طراحی و پیاده‌سازی سیستم تشخیص خودکار آریتمی و همین‌طور نتایج به دست آمده را شرح خواهیم داد.

^{۱۰}Fuzzy systems

^{۱۱}Artificial neural networks

^{۱۲}Support Vector Machines (SVM)

فصل ۲

مفاهيم اوليه

۱-۲ قلب و نحوه عملکرد آن

قلب ماهیچه‌ای متشکل از ۴ حفره است. دو حفره بالایی، دهلیزهای چپ و راست نامیده می‌شوند و دو حفره پایینی، بطن‌های چپ و راست نام دارند. در هر سیکل تپش قلب، خون بدون اکسیژن از طریق بزرگ‌سیاهرگ‌های بالایی و پایینی وارد دهلیز راست می‌شود. پس از طی فرایندی در قلب، خون دارای اکسیژن شده و از بطن چپ خارج می‌شود. این خون سپس از طریق سرخرگ‌ها به اعضای بدن می‌رسد. قلب یک فرد بزرگسال سالم، به طور متوسط بین ۶۰ تا ۱۰۰ بار در دقیقه می‌تپد. [۱۶]

عملکرد قلب توسط یک سیستم الکتریکی و به وسیله سیگنال‌های تولید شده در آن کنترل می‌شود. این سیگنال‌ها دیواره‌های قلب را تحریک می‌کنند و با انقباض دیواره‌ها، خون از قلب خارج شده و در سیستم گردش خون جریان می‌یابد. در ادامه به طور دقیق به نحوه عملکرد قلب می‌پردازیم.

۱-۱-۲ سیستم هدایت الکتریکی قلب

تمامی فعالیت‌های قلب که منجر به پمپ‌کردن خون در بدن می‌شوند، تحت کنترل سیستم هدایت الکتریکی قلب^۱ قرار دارند. این سیستم با انتقال الکتریکی سیگنال‌های تولید شده، باعث به تپش درآمدن ماهیچه‌ای قلب می‌شود. بخش‌های اصلی این سیستم عبارت اند از:

- ۱- گره سینوسی‌دهلیزی^۲ (SA) در دهلیز راست قلب
 - ۲- گره دهلیزی‌بطنی^۳ (AV) در سپتوم داخل‌دهلیزی قلب^۴ (دیواره‌ای ماهیچه‌ای که دهلیز راست و چپ قلب را جدا می‌کند)
 - ۳- سیستم هیس-پورکینژ^۵ در دیواره‌های بطن‌های قلب
- این بخش‌ها در شکل ۱-۲ قابل مشاهده هستند.

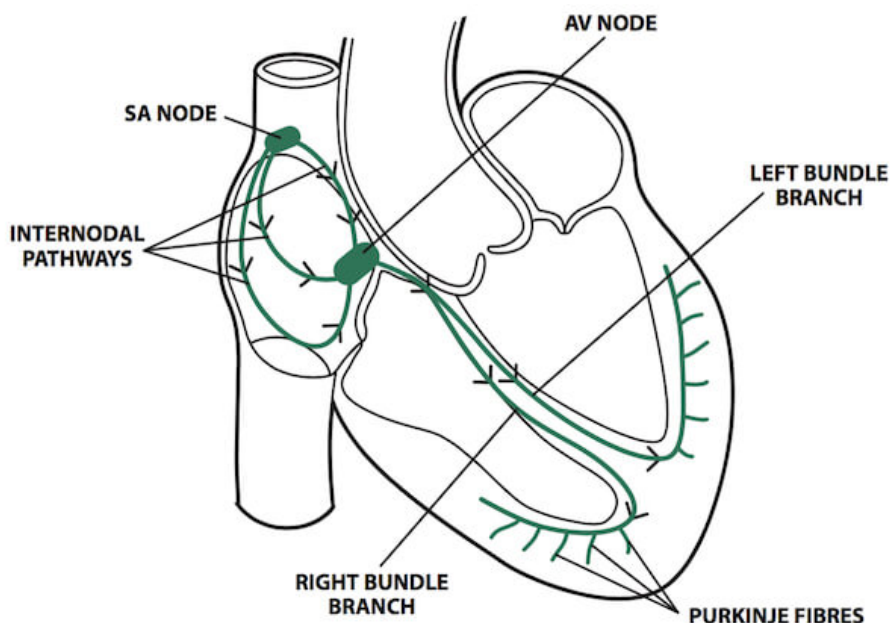
^۱Cardiac conduction system

^۲Sinoatrial node

^۳Atrioventricular node

^۴Interatrial septum

^۵His-Purkinje system



شکل ۱-۲ سیستم هدایت الکتریکی قلب [۲]

نقطه‌ی آغاز هر ضربان قلب، گره سینوسی‌دهلیزی است. این گره با تولید سیگنالی هر دو دهلیز را تحریک به انقباض می‌کند و در نتیجه‌ی این عمل، خون از طریق دریچه‌های باز، از دو دهلیز وارد دو بطن قلب می‌شود. سپس سیگنال وارد گره دهلیزی‌بطنی شده و برای لحظه‌ای کوتاه تاخیر می‌کند، تا خون فرصت پر کردن دو بطن قلب را پیدا کند. در مرحله‌ی بعد، سیگنال آزاد شده و در مسیری به نام دسته‌ی هیس^۶ واقع در دیواره‌های بطن‌ها حرکت خود را ادامه می‌دهد. در این مرحله، سیگنال به دو دسته تقسیم شده و این دو دسته از طریق دو مسیر به نام‌های فیبرهای پورکینژ^۷ چپ و راست، به ترتیب وارد بطن چپ و راست قلب می‌شوند. این عمل باعث انقباض دو بطن می‌شود و در نتیجه‌ی این عمل، خون از طریق دریچه‌های بیرونی قلب، از آن خارج شده و به ریه‌ها و بقیه‌ی اعضای بدن انتقال می‌یابد. در این مرحله سیگنال از بطن‌ها گذر می‌کند و دو بطن وارد حالت استراحت می‌شوند، تا سیگنال بعدی فرا برسد.

تولید پی‌درپی این سیگنال‌ها، باعث انقباض و استراحت منظم و هماهنگ قلب شده و ضربان قلب را ایجاد می‌کند. در واقع ضربان قلب هر شخص، توسط تعداد دفعاتی در طول یک دقیقه که گره سینوسی‌دهلیزی سیگنال تولید می‌کند تعیین می‌شود. [۱۷]

^۶Hiss bundle

^۷Purkinje fibers

۲-۲ آریتمی قلبی

آریتمی قلبی به دسته‌ای از بیماری‌های قلبی اطلاق می‌شود که در آن‌ها، آهنگ تپش قلب حالتی غیرعادی پیدا می‌کند. به طور کلی دلیل رخ دادن آریتمی، عدم انتقال درست سیگنال‌های الکتریکی قلب بیان می‌شود. تعدادی از انواع آریتمی‌ها می‌توانند شدیداً خطرناک و کشنده باشند. اکثر آریتمی‌ها بی خطر شناخته شده‌اند، اما در صورت عدم تشخیص و رسیدگی به موقع می‌توانند زندگی عادی فرد مبتلا را آشفته‌ساخته یا حیات او را تهدید کنند.

۱-۲-۲ انواع آریتمی قلبی

آریتمی‌ها بر اساس نوع اختلالی که در ضربان قلب ایجاد می‌کنند، به چهار دسته‌ی کلی تقسیم می‌شوند.

۱- ضربان‌های زودرس^۸: در این دسته از آریتمی‌ها، قلب ضربان‌هایی زودرس تولید می‌کند که آهنگ طبیعی تپش آن را مختل می‌کنند. در صورتی که ضربان زودرس در بطن قلب تولید شده‌باشد، ضربان زودرس بطنی^۹، و در صورتی که در دهلیز ایجاد شده باشد، ضربان زودرس دهلیزی^{۱۰} نامیده می‌شود.

۲- تاکی کاردی فوق بطنی^{۱۱}: در این نوع آریتمی، قلب به صورتی غیرعادی تندتر از معمول (تقریباً بیش از ۱۰۰ ضربان در دقیقه) می‌تپد. [۱۸] این آریتمی‌ها در بین گره سینوسی‌دهلیزی و گره دهلیزی‌بطنی ایجاد می‌شوند.

۳- آریتمی‌های بطنی^{۱۲}: آریتمی‌هایی که از پایین گره دهلیزی‌بطنی (در سطح بطن قلب) ریشه می‌گیرند در این دسته قرار دارند.

۴- برادی کاردی^{۱۳}: در این نوع آریتمی، قلب بیمار آرام‌تر از حالت عادی می‌تپد و نرخ ضربان قلب معمولاً پایین‌تر از ۶۰ تپش در دقیقه است. [۱۹]

^۸Premature beats

^۹Premature Ventricular Complex (PVC)

^{۱۰}Premature Atrial Complex (AVC)

^{۱۱}Supraventricular Tachycardia (SVT)

^{۱۲}Ventricular arrhythmia

^{۱۳}Bradycardia

۳-۲ سیگنال نوار قلب

همان طور که گفته شد، سلول‌های گره سینوسی تحریک الکتریکی منظمی را ایجاد می‌کنند که توسط سیستم هدایت الکتریکی موجود در قلب، به بخش‌های دیگر آن انتشار یافته و باعث تپش متناوب قلب می‌شود. نتیجه‌ی این فعالیت، ایجاد جریان الکتریکی در سطح بدن و تحریک تغییرات در پتانسیل الکتریکی سطح پوست است. این سیگنال‌ها را می‌توان به وسیله‌ی الکترودها و دیگر تجهیزات، ثبت و اندازه‌گیری نمود.

در فرایند ثبت نوار قلب، اختلاف پتانسیل بین نقاط قرارگیری الکترودها بر روی بدن اندازه‌گیری شده و معمولاً به کمک تقویت‌کننده‌های عملیاتی^{۱۴} بهبود داده می‌شود. در مرحله‌ی بعد، سیگنال ابتدا از یک فیلتر بالاگذر و سپس از یک فیلتر پایین‌گذر تصحیح فرکانس عبور داده می‌شود. در نهایت این سیگنال آنالوگ، به سیگنال دیجیتال تبدیل می‌شود. منحنی گرافیکی رسم شده در انتهای این فرایند، نوار قلب، و یا به اختصار ECG نامیده می‌شود.

امروزه در روش‌های استاندارد اندازه‌گیری نوار قلب، تعدادی الکترودها بر روی سطح پوست قرار می‌گیرند و یکی از آن‌ها به عنوان مرجع^{۱۵} برای دیگر الکترودها در نظر گرفته می‌شود. به طور معمول، الکترودها مرجع روی ساق پای راست نصب می‌شود. [۲۰] هر یک از الکترودهای دیگر، ولتاژ ناحیه‌ی قرارگیری خود را نسبت به ولتاژ الکترودها مرجع اندازه‌گیری می‌کنند. هر یک از این اختلاف پتانسیل‌های اندازه‌گیری شده، یک لید^{۱۶} نامیده می‌شود.

۱-۳-۲ نحوه‌ی قرارگیری الکترودها بر روی پوست و لیدهای تولیدشده

یکی از ترکیب‌های رایج قراردادن الکترودها متشکل از ۱۰ الکترودها است که بر روی دست، پا و سینه‌ی بیمار قرار می‌گیرند. از ترکیب این الکترودها ۱۲ لید ایجاد می‌شود که به سه دسته‌ی کلی تقسیم می‌شوند:

• سه لید دوقطبی اندامی^{۱۷} به نام‌های I، II و III

• سه لید تک‌قطبی اندامی^{۱۸} به نام‌های aVR، aVL و aVF

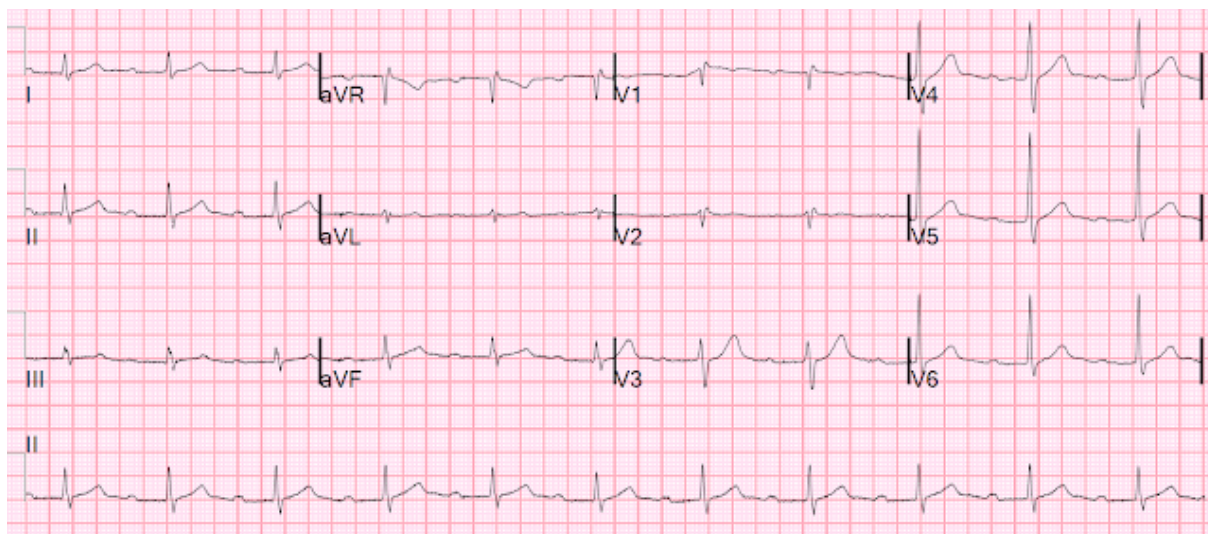
^{۱۴}Operational amplifiers

^{۱۵}Reference

^{۱۶}Lead

^{۱۷}Bipolar limb leads

^{۱۸}Unipolar limb leads



شکل ۲-۲ نوار قلب ۱۲ لیدی گرفته شده از یک فرد سالم [۳]

• شش لید تک قطبی سینه‌ای^{۱۹} به نام‌های V1 تا V6

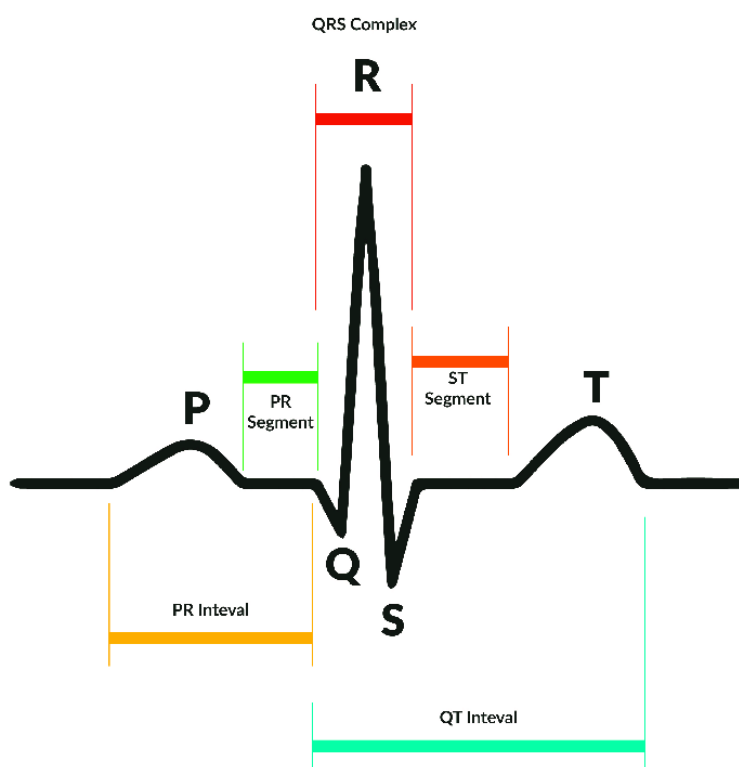
هر یک از این لیدها فعالیت الکتریکی قلب را از یک زاویه‌ی خاص در بدن نشان می‌دهد. پرکاربردترین لید برای تشخیص بیماری‌های قلبی، لید II می‌باشد که اختلاف پتانسیل بین الکترودهای ساق پای چپ و بازوی راست را نشان می‌دهد. در شکل ۲-۲ یک نوار قلب ۱۲ لیدی مشاهده می‌شود. منحنی رسم شده از هر لید به صورت جداگانه نشان داده شده‌است و لید II نیز به تنهایی رسم شده‌است. این لید به خصوص از آن جهت اهمیت دارد که نمای خوبی از ترکیب QRS ارائه می‌دهد. در بخش بعد در مورد این موضوع به تفصیل توضیح داده خواهد شد.

۲-۳-۲ ترکیب QRS

با بررسی یک سیکل ضربان قلب در نوار قلب، ۵ انحراف^{۲۰} یا موج پراهمیت دیده می‌شود. اولین موج، P نام دارد که با فعال شدن دهلیزهای راست و چپ و بالارفتن پتانسیل الکتریکی آن‌ها اتفاق می‌افتد. سه موج بعدی به ترتیب Q، R و S نام دارند. این سه موج به ترتیب و با فاصله‌ی کمی از هم رخ می‌دهند و عموماً به عنوان یک ترکیب، همراه یکدیگر بررسی می‌شوند. این ترکیب که QRS نامیده می‌شود، واضح‌ترین بخش مشاهده شده در یک سیکل قلبی است که مدت زمان بالارفتن پتانسیل ماهیچه‌های بطنی قلب را نشان می‌دهد. موج بعدی T نام دارد که در طول آن بطن‌ها منقبض شده و بار

^{۱۹}Unipolar chest leads

^{۲۰}Deflection



شکل ۲-۳ ترکیب QRS [۴]

مثبت خود را تخلیه می‌کنند. ترکیب QRS در شکل ۲-۳ مشاهده می‌شود.

۱-۲-۳-۲ بازه‌های زمانی مهم در سیکل ضربان قلب

مهم‌ترین بازه‌های زمانی در یک سیکل ضربان قلب عبارت اند از:

- بازه‌ی PR: فاصله‌ی زمانی از ابتدای موج P تا ابتدای ترکیب QRS
- مدت‌زمان QRS: مدت‌زمان رخدادن ترکیب QRS
- بازه‌ی QT: فاصله‌ی زمانی از ابتدای ترکیب QRS تا انتهای موج T
- بازه‌ی RR: مدت‌زمان سیکل کامل قلب که نشان‌دهنده‌ی سیکل کامل بطن‌ها می‌باشد.
- بازه‌ی PP: مدت‌زمان سیکل کامل دهلیزی

۲-۳-۲ تاثیر آریتمی قلبی بر روی شکل ترکیب QRS

وجود آریتمی قلبی می‌تواند باعث تغییر شدید در امواج Q، R و S شود. لید II به دلیل واضح‌تر نشان دادن ترکیب QRS و لیدهای V1 تا V6 به دلیل این که الکترودهای آن‌ها بر روی سینه قرار گرفته و تشخیص بهتر تغییرات پتانسیل ماهیچه‌ی بطنی را ممکن می‌سازند، تا کنون بهترین نتایج را در تشخیص آریتمی نشان داده‌اند. [۲۰]

در طول بازه‌ی زمانی QRS بطن‌ها به وسیله‌ی سیستم هیس-پورکینژ منقبض می‌شوند. این سیستم شامل سلول‌هایی در دیواره‌های بطن‌ها است که خاصیت رسانایی سریع الکتریکی را دارند. در صورت ایجاد اختلال در کار این سیستم و ضعیف‌شدن خاصیت رسانایی الکتریکی سلول‌ها، بازه‌ی زمانی QRS طولانی‌تر می‌شود. در برخی موارد سیگنال الکتریکی به جای انتقال یافتن از طریق سیستم هیس-پورکینژ، از طریق ماهیچه‌های قلب منتقل می‌شود. این اتفاق منجر به طولانی شدن زمان انتقال الکتریکی سیگنال و در نتیجه عریض شدن بازه‌ی QRS می‌شود. به طور معمول طول یک بازه‌ی QRS بین ۰/۰۸ تا ۰/۱ ثانیه است. در مواردی که طول این بازه از ۰/۱۲ ثانیه بیشتر شود، QRS غیرعادی تلقی می‌شود. [۲۱]

۴-۲ مسائل دسته‌بندی

در مسائل دسته‌بندی، ورودی‌های مسئله تعدادی داده هستند و مطلوب مسئله، جای دادن هر یک از داده‌ها در یک دسته یا کلاس است. به بیان رسمی‌تر در این مسئله‌ها، هدف، تخمین‌زدن یک نگاشت از متغیرهای ورودی X به تعدادی متغیر خروجی گسسته Y است. این متغیرهای خروجی تعدادی برجسب^{۲۱} هستند که تعیین می‌کنند هر داده در کدام دسته قرار می‌گیرد. تعداد این دسته‌ها می‌تواند دو و یا بیشتر باشد که در حالت دوم، مسئله یک مسئله‌ی دسته‌بندی چنددسته‌ای^{۲۲} نامیده می‌شود. در ادامه، یکی از روش‌های دسته‌بندی به نام روش ماشین بردار پشتیبانی که در این پروژه مورد استفاده قرار گرفته است، توضیح داده می‌شود.

^{۲۱}Label

^{۲۲}Multiclass classification problem

۱-۴-۲ روش ماشین بردار پشتیبانی (SVM)

یکی از پرکاربردترین الگوریتم‌ها برای حل مسائل دسته‌بندی، الگوریتم‌های SVM هستند. در این الگوریتم‌ها، داده‌ها به مثابه‌ی نقطه‌هایی در یک فضای N بعدی فرض می‌شوند. هدف الگوریتم، یافتن ابرصفحه‌هایی^{۲۳} است که به طور بهینه نقطه‌های داده‌ها را به کلاس‌های متعدد دسته‌بندی کند. تعداد بعدهای این فضا (N) برابر با تعداد ویژگی‌ها است. معمولاً تعداد زیادی ابرصفحه را می‌توان برای جداسازی دو کلاس مختلف از داده‌ها یافت، اما در این الگوریتم، هدف یافتن ابرصفحه‌ای است که بیشترین فاصله را با نزدیک‌ترین نقطه‌ی داده در هر یک از کلاس‌ها داشته‌باشد. این فاصله، حاشیه^{۲۴} نامیده می‌شود.

۱-۴-۲-۱ ابرصفحه

ابرفصفحه مرزی است که نقاط داده‌ها را در یک فضای N بعدی به دو بخش تقسیم می‌کند. برای مثال در مسئله‌ای با دو کلاس هدف، نقاطی که در هر یک از دو سمت ابرصفحه‌ی به دست آمده قرار می‌گیرند، به یکی از آن دو کلاس تعلق می‌یابند. تعداد بعد ابرصفحه بسته به تعداد ویژگی‌های داده‌ها است. مثلاً در مسئله‌ای که سه ویژگی برای داده‌ها به دست آورده‌ایم، فضای داده ۳ بعدی بوده و در نتیجه ابرصفحه‌ی جداکننده‌ی داده‌ها نیز ۳ بعدی خواهد بود.

۱-۴-۲-۲ بردار پشتیبانی

بردارهای پشتیبانی، نقاط داده‌ای هستند که ابرصفحه را تعریف می‌کنند. این نقاط به ابرصفحه نزدیک‌تر بوده و بر روی موقعیت قرارگیری و جهت آن تاثیر می‌گذارند. به کمک این بردارها، ابرصفحه‌ای با بیشترین حاشیه برای دسته‌بندی انتخاب می‌شود. [۵] نمودار یک مسئله‌ی دسته‌بندی دوبعدی در شکل ۴-۲ دیده می‌شود.

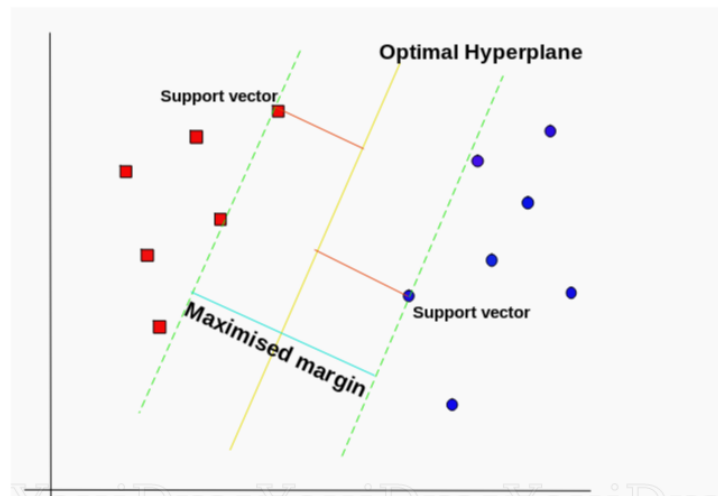
۱-۴-۲-۳ تابع کرنل

در روش SVM برای دسته‌بندی داده‌ها از توابعی به نام توابع کرنل استفاده می‌شود. تابع کرنل داده را به عنوان ورودی گرفته و آن را به فضایی دیگر انتقال^{۲۵} می‌دهد. به کمک تابع کرنل، داده‌هایی که در فضای عادی مشاهده شده‌اند، به فضایی

^{۲۳}Hyperplanes

^{۲۴}Margin

^{۲۵}Transform



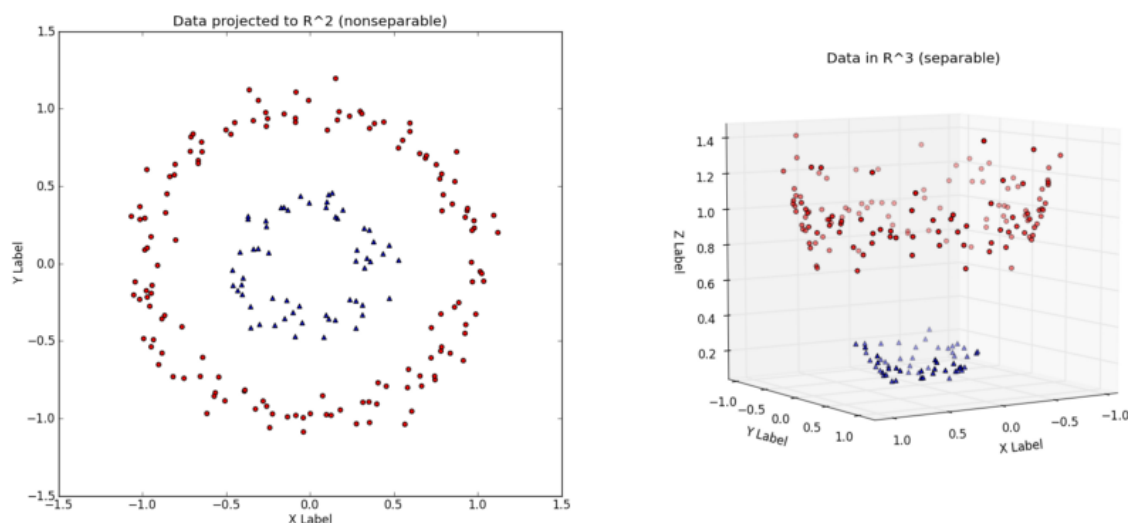
شکل ۲-۴ نموداری از حل یک مسئله‌ی دسته‌بندی دوبعدی با روش SVM [۵]

با تعداد ابعاد بالاتر انتقال می‌یابند که در چنین فضایی امکان جداسازی آن‌ها وجود دارد. در واقع هر مدل خطی را می‌توان به کمک تابع کرنل به یک مدل غیر خطی تبدیل کرد، به این صورت که ویژگی‌های مدل را با یک تابع کرنل جایگزین کنیم. به طور رسمی‌تر می‌توان تابع کرنل را به این صورت تعریف کرد: به ازای هر x_i و x_j در فضای X می‌توان توابعی به صورت $k(x_i, x_j)$ را یافت که حاصل ضرب داخلی دو نقطه در فضای دیگری به نام V هستند. این روابط در معادله‌ی ۱-۲ قابل مشاهده است. [۲۲] در این جا Φ نگاشتی از ویژگی‌های فضای X به V است و نقاط $\langle \Phi(X_i), \Phi(X_j) \rangle$ حاصل انجام این نگاشت بر روی نقطه‌ی (x_i, x_j) هستند.

$$k : X \times X \rightarrow \mathbb{R} \quad (1-2)$$

$$k(x_i, x_j) = \langle \Phi(X_i), \Phi(X_j) \rangle$$

ساده‌ترین نوع کرنل، کرنل خطی است. این توابع داده‌ها را به فضایی با تعداد بعد بالاتر نگاشت نمی‌کنند، به همین دلیل بهتر است در مسائلی که داده‌ها به صورت خطی قابل جداسازی هستند، از این نوع کرنل استفاده شود. این نوع کرنل‌ها به دلیل سادگی و خطی بودن، سرعت بیشتری در دسته‌بندی دارند. معمولاً در مسائلی که تعداد ویژگی‌ها زیاد بوده و نگاشت داده‌ها به نقاطی در فضای با تعداد بعدهای بالاتر تاثیر چشمگیری در بهبود دسته‌بندی ندارد، از کرنل خطی استفاده می‌شود. [۲۳]



شکل ۲-۵ سمت چپ: داده‌های غیر قابل جداسازی توسط یک ابرصفحه در یک فضای دوبعدی، سمت راست: داده‌های انتقال داده‌شده به فضای سه‌بعدی و قابل جداسازی [۶]

نوع پیچیده‌تری از کرنل که در بسیاری از مسائل دسته‌بندی کاربرد دارد، کرنل RBF^{۲۶} نام دارد. این تابع بر روی

نقطه‌ی X_i و X_j در فضای X که یک فضای ورودی است در معادله‌ی ۲-۲ قابل مشاهده است. [۲۴]

$$k(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right) = \exp(-\gamma\|X_i - X_j\|^2) \quad (2-2)$$

در این رابطه σ یک پارامتر آزاد است. اعمال این تابع باعث می‌شود نقاط داده به نقاطی در فضایی با تعداد بعد بیشتر نگاشت شوند. در مسائلی که در فضای اصلی داده‌های ورودی، ابرصفحه‌ای برای جداسازی کلاس‌ها یافت نمی‌شود، می‌توان با استفاده از کرنل RBF در فضایی با تعداد بعد بالاتر، ابرصفحه‌ای برای جداسازی کلاس‌ها یافت. این موضوع در شکل ۲-۵ قابل مشاهده است. این نوع تابع کرنل، زمان و توان پردازشی بیشتری به نسبت کرنل خطی نیاز دارد.

۲-۴-۱-۴ انجام دسته‌بندی با استفاده از تابع کرنل RBF

برای انجام عمل دسته‌بندی با استفاده از کرنل RBF، لازم است تعدادی پارامتر برای این تابع تعیین شوند. میزان تاثیر هر یک از این پارامترها بر روی نتیجه‌ی نهایی دسته‌بندی معمولاً به کاربرد وابسته است. در ادامه تعدادی از مهم‌ترین

^{۲۶} Radial Basis Function

پارامترها توضیح داده می شوند.

• پارامتر γ :

γ پارامتر آزادی است که در تابع کرنل RBF وجود دارد. این پارامتر تعیین می کند یک داده به تنهایی چقدر می تواند بر روی نتیجه ی نهایی دسته بندی تاثیر داشته باشد. در صورت کوچک بودن γ این تاثیر زیاد و در صورت بزرگ بودن آن، این تاثیر کم است. این پارامتر را می توان به صورت عکس شعاع تاثیر نمونه هایی که مدل به عنوان بردار پشتیبانی انتخاب می کند دانست. گامای کوچک باعث می شود منحنی گاوسی تابع کرنل، واریانس زیادی داشته باشد. اگر X_j یک بردار پشتیبانی باشد، کوچک بودن γ نتیجه می دهد که کلاس این بردار پشتیبانی، بر روی تشخیص کلاس X_i تاثیر دارد حتی اگر فاصله ی آن ها زیاد باشد. برعکس اگر γ بزرگ باشد، واریانس کوچک بوده و این نتیجه می دهد یک بردار پشتیبانی تاثیر زیادی بر روی تشخیص کلاس نمونه ها ندارد.

رفتار مدل نسبت به مقدار γ بسیار حساس است. به طور کلی می توان گفت بزرگ بودن بیش از حد γ باعث می شود شعاع ناحیه ای که بردار پشتیبانی بر روی آن تاثیر دارد بسیار کوچک شده و تنها خود بردار را در بر بگیرد. کوچک بودن بیش از حد آن نیز باعث می شود ناحیه ی تاثیر هر یک از بردارهای پشتیبانی به اندازه ی کل مجموعه ی داده ها بزرگ می شود و مدل نهایی تفاوتی با یک کرنل خطی که در آن تعدادی ابرصفحه نقاط داده را از هم جدا می کنند نخواهد داشت.

• پارامتر C :

در SVM هدف پیدا کردن مرز جداکننده ای است که تمامی داده های مربوط به هر یک از کلاس ها را به درستی جدا کند. در صورت وجود خطا در نمونه ها و یا داده های غیرعادی، این کار باعث می شود مدل نتواند مرز مناسبی برای جداسازی کلاس ها بیابد. به همین علت مفهوم حاشیه ی نرم^{۲۷} مطرح می شود. با اعمال حاشیه ی نرم، به SVM اجازه داده می شود برخی از نمونه ها را در دسته بندی در نظر نگیرد و برخی از نمونه ها را در کلاس نادرست دسته بندی کند. پارامتر C شدت این عمل را کنترل می کند. این پارامتر تاثیر هر یک از بردارهای پشتیبانی بر روی حاشیه ی ابرصفحه ی جدا کننده را نشان می دهد. مدلی با C پایین تر، آسان گیرانه تر دسته بندی کرده و منجر به

^{۲۷}Soft margin

داشتن داده‌های بیشتری با دسته‌بندی نادرست می‌شود، اما در عوض حاشیه‌ی بالاتری را نتیجه می‌دهد.

۲-۴-۲ دسته‌بندی داده‌ها با استفاده از روش SVM

روش کلی ساخت یک مدل SVM به این صورت است که داده‌ها را به دو مجموعه‌ی داده‌های آموزشی^{۲۸} و داده‌های تست^{۲۹} تقسیم می‌کنیم. نحوه‌ی تقسیم داده‌ها به این دو مجموعه تا حد زیادی به مسئله وابسته است. نحوه‌ی کلی انجام دسته‌بندی به این صورت است که ابتدا عملیات آموزش بر روی مجموعه‌ی اول انجام شده و مدل SVM ساخته می‌شود. سپس این مدل بر روی مجموعه‌ی تست آزموده شده و دقت دسته‌بندی، با توجه به معیارهای کارایی مورد نظر در مسئله اندازه‌گیری می‌شود.

۱-۲-۴-۲ تعیین پارامترهای مدل با استفاده از روش اعتبارسنجی متقابل

در هنگام ساخت مدل SVM پارامترهایی مانند پارامتر C باید به صورت دستی برای مدل تعیین شوند. ممکن است مقداری که برای یک پارامتر در مدل تعیین می‌شود، به نحوی دست‌کاری شود که مدل به‌دست‌آمده دقت خوبی در دسته‌بندی داده‌های تست داشته‌باشد، اما بر روی داده‌های ناشناخته‌ی دیگر این دقت را تضمین نکند. در چنین حالتی پارامترها تنها به هدف بالابردن دقت دسته‌بندی در داده‌های مجموعه‌ی تست تعیین شده‌اند و در واقع مدل، دانشی از مجموعه‌ی تست دارد. ارزیابی مدل در این شرایط نمی‌تواند مفید باشد، چرا که میزان موفقیت آن در دسته‌بندی داده‌های جدیدی که هیچ پیش‌فرضی از آن‌ها ندارد، پایین‌تر از مقدار به‌دست‌آمده خواهد بود.

در این شرایط، از روشی به نام اعتبارسنجی متقابل^{۳۰} برای تعیین پارامترهای الگوریتم، به خصوص C استفاده می‌شود. در اعتبارسنجی متقابل k لایه‌ای^{۳۱}، مجموعه‌ی آموزش به k بخش تقسیم می‌شود، سپس مدلی با استفاده از داده‌های $k-1$ بخش از این بخش‌ها آموزش داده شده و بر روی یک بخش باقی‌مانده آزموده می‌شود. این عمل k بار برای هر k بخش موجود اجرا خواهد شد. در انتها معیارهای کارایی گزارش شده توسط این روش، میانگین معیارهای کارایی به‌دست‌آمده در هر یک از لایه‌ها خواهد بود. این عمل برای مقادیر مختلفی از پارامتر مورد نظر، مثلاً C ، اجرا شده و مقداری که بهترین معیارهای کارایی را نتیجه دهد، به عنوان مقدار پارامتر تعیین می‌شود.

^{۲۸} Training data set

^{۲۹} Test data set

^{۳۰} Cross-validation

^{۳۱} K-fold cross-validation

۵-۲ معیارهای کارایی در مسائل دسته‌بندی

پس از استخراج ویژگی‌ها و آموزش مدل دسته‌بندی، باید میزان موفقیت الگوریتم در دسته‌بندی داده‌های تست سنجیده شود. برای این منظور، معیارهای استاندارد تعریف شده‌اند. پیش از پرداختن به این معیارها لازم است با مقادیر مثبت و منفی صحیح و کاذب در مسائل دسته‌بندی چنددسته‌ای آشنا شویم.

در یک مسئله دسته‌بندی چند دسته‌ای مقادیر زیر برای هر کلاس تعریف می‌شوند:

- مثبت صحیح^{۲۲} (TP): داده‌هایی که به درستی در این کلاس دسته‌بندی شده‌اند.
- مثبت کاذب^{۲۳} (FP): داده‌هایی که در دسته‌بندی متعلق به این کلاس تشخیص داده شده‌اند اما در واقعیت به کلاس دیگری تعلق دارند.
- منفی صحیح^{۲۴} (TN): داده‌هایی که عدم تعلق آن‌ها به این کلاس به درستی تشخیص داده شده است.
- منفی کاذب^{۲۵} (FN): داده‌هایی که الگوریتم آن‌ها را متعلق به کلاس‌های دیگر تشخیص داده‌است، اما در واقع به این کلاس تعلق دارند.

۱-۵-۲ ماتریس درهم‌ریختگی

ماتریس درهم‌ریختگی^{۲۶} یک ماتریس مربعی است که برای مشاهده و ارزیابی نتایج الگوریتم‌های دسته‌بندی به کار می‌رود. تعداد سطرها و ستون‌های این ماتریس به تعداد کلاس‌های هدف است. یکی از محورهای ماتریس، برچسب‌های پیش‌بینی شده توسط مدل برای داده‌های مجموعه‌ی تست و محور دیگر، برچسب‌های واقعی آن داده‌ها را نشان می‌دهد. در شکل ۶-۲ ماتریس درهم‌ریختگی یک مسئله دسته‌بندی چنددسته‌ای را مشاهده می‌کنیم. درایه‌ی $[x, y]$ در این ماتریس نشان می‌دهد چه تعداد داده وجود دارد که به وسیله‌ی الگوریتم آن‌ها در دسته‌ی x جای داده شده‌اند، اما در واقع به دسته‌ی y تعلق دارند. درایه‌های قطری نشان‌دهنده‌ی مثبت‌های صحیح برای هر کلاس هستند. از این ماتریس می‌توان

^{۲۲}True Positive

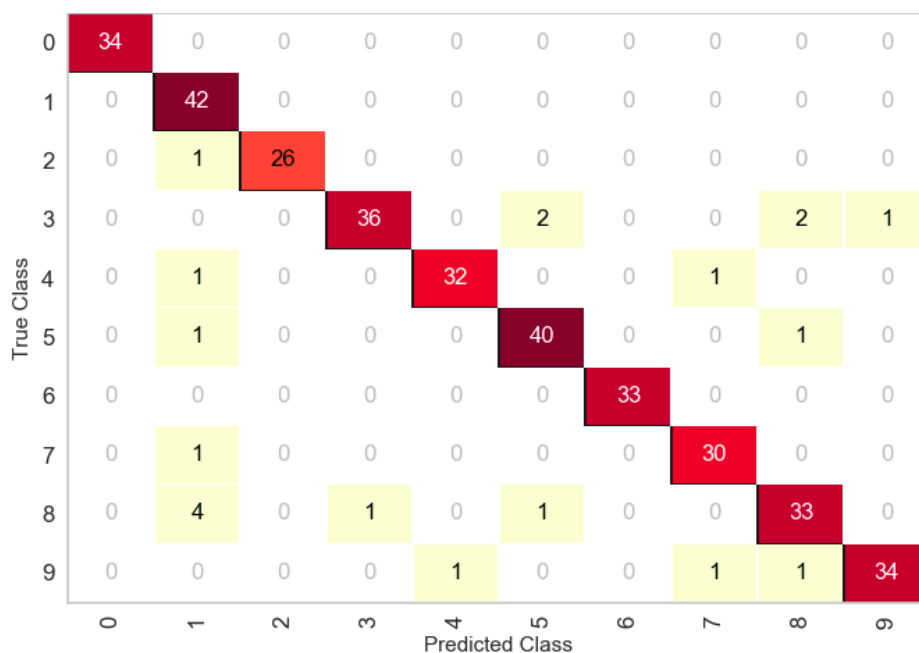
^{۲۳}False Positive

^{۲۴}True Negative

^{۲۵}False Negative

^{۲۶}Confusion matrix

اطلاعات مفیدی برای ارزیابی مدل‌های دسته‌بندی استخراج نمود.



شکل ۲-۶ ماتریس درهم‌ریختگی برای یک مسئله دسته‌بندی چنددسته‌ای [۷]

۲-۵-۲ دقت و حساسیت

در یک مسئله دسته‌بندی چند دسته‌ای، برای هر دسته مقادیر دقت^{۳۷} و حساسیت^{۳۸} تعریف می‌شوند. دقت دسته‌بندی برای کلاس A (که نرخ پیش‌بینی‌پذیری مثبت^{۳۹} نیز نامیده می‌شود) نشان می‌دهد چقدر احتمال دارد که داده‌ای که در A دسته‌بندی شده است، واقعا به این دسته تعلق داشته باشد. رابطه‌ی ۲-۳ نحوه‌ی محاسبه‌ی دقت را نشان می‌دهد.

$$Precision = \frac{TP}{TP + FP} \quad (۳-۲)$$

^{۳۷}Precision

^{۳۸}Sensitivity

^{۳۹}Positive predictivity

حساسیت محاسبه شده برای کلاس A نشان می دهد چقدر احتمال دارد داده ای که در واقع به کلاس A تعلق دارد، در این کلاس دسته بندی شود. رابطه ی ۴-۲ نحوه ی محاسبه ی این مقدار را نشان می دهد.

$$Sensitivity = \frac{TP}{TP + FN} \quad (۴-۲)$$

۳-۵-۲ صحت و صحت کلی

صحت^{۴۰} به صورت رابطه ی ۵-۲ تعریف می شود. این مقدار را می توان برای هر کلاس به صورت جدا محاسبه کرد، که در این صورت نسبت مجموع داده هایی که به درستی متعلق و یا غیر متعلق به یک کلاس تشخیص داده شده اند، به کل داده هایی که در آن کلاس دسته بندی شده اند را نشان می دهد. صحت را می توان برای تمامی کلاس ها نیز محاسبه کرد، که در این صورت به آن صحت کلی^{۴۱} گفته می شود. صحت کلی، نسبت داده هایی که به درستی در تمام دسته ها دسته بندی شده اند به کل داده های تست را نشان می دهد.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (۵-۲)$$

^{۴۰} Accuracy

^{۴۱} Overall accuracy

فصل ۳

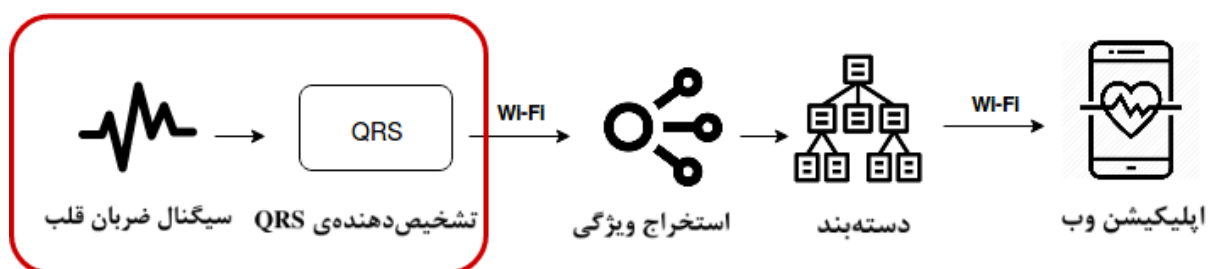
روش حل مسئله

۱-۳ مقدمه

این پروژه در دو بخش کلی پیش‌پردازش در سمت سخت‌افزار و پردازش در سرور انجام شده‌است. در بخش اول، تعدادی پردازش اولیه بر روی داده‌های خام ضربان قلب انجام می‌شود. این بخش یک بستر پیاده شده بر روی سخت‌افزار است که برای کامل شدن باید به یک سنسور ضربان قلب متصل شود. این بخش همراه بیمار خواهد بود و پردازش‌های ساده‌ی اولیه را بر روی سیگنال نوار قلب انجام داده و نتایج را به سرور ارسال می‌کند. پردازش‌های پیچیده‌تر برای تشخیص آریتمی بر عهده‌ی سرور خواهد بود. در سرور یک الگوریتم دسته‌بندی بر روی داده‌ها انجام شده و کلاس آریتمی آن‌ها تشخیص داده می‌شود.

۲-۳ عملیات پیش‌پردازش بر روی سخت‌افزار

در این بخش عملیات پیش‌پردازش با هدف تشخیص ترکیب QRS در هر ضربان قلب بر روی سیگنال دیجیتال ضربان قلب اجرا می‌شود. خروجی این عملیات، موقعیت زمانی قله‌ی R در ترکیب QRS هر ضربان است که در پردازش‌های آینده برای تشخیص آریتمی آن ضربان مورد استفاده قرار می‌گیرد. این بخش بر روی یک ماژول ESP8266 که یک ماژول وای‌فای دارای یک پردازنده‌ی ۸۰ مگاهرتزی است، پیاده‌سازی می‌شود. این ماژول علاوه بر انجام پیش‌پردازش، ارسال داده‌های حاصل از آن از طریق وای‌فای را نیز بر عهده دارد. قسمت‌های پیاده‌سازی شده در این بخش در شکل ۱-۳ در کادر مشخص شده‌اند.



شکل ۱-۳ بخش پیش‌پردازش بر روی سخت‌افزار در مراحل تشخیص آریتمی

۱-۲-۳ مراحل تشخیص QRS

پیش از ورود سیگنال نوار قلب به ماژول پیش پردازش، نوار قلب خام گرفته شده از بیمار از یک مبدل آنالوگ به دیجیتال^۱ عبور کرده و با نرخ نمونه برداری^۲ معینی به سیگنال دیجیتال تبدیل می شود. مقدار این نرخ نمونه برداری در برخی مراحل پیش پردازش اهمیت دارد. پس از دیجیتال شدن، سیگنال وارد ماژولی که برای تشخیص QRS طراحی کرده ایم می شود. در ادامه به مراحل اصلی طی شده در این بخش می پردازیم.

۱-۱-۲-۳ حذف نویز سیگنال به کمک فیلتر میان گذر

اولین مرحله در تشخیص QRS حذف نویز سیگنال نوار قلب است. در حین ثبت ضربان قلب، منابع مختلفی از نویز در سیگنال اختلال ایجاد می کنند. در یک سیگنال ECG به طور معمول نویزهای فرکانس پایینی ناشی از نوسانات الکتروود مرجع وجود دارد. این نویزها به علت حرکت الکتروودها بر روی پوست و همین طور اعمالی چون حرکات و تنفس بیمار به وجود می آیند. انقباض ماهیچه های اطراف قلب نیز یکی دیگر از منابع نویز است. این انقباضات توسط الکتروودها ثبت شده و در نوار قلب نویزهای فرکانس بالایی ایجاد می کنند. [۲۵]

با توجه به نویزهای معمول، محدوده ی فرکانسی مطلوب برای بیشینه کردن انرژی QRS و کمینه کردن انرژی نویز، ۵ تا ۱۵ هرتز تشخیص داده شده است. [۲۶] به منظور نگه داشتن این بازه ی فرکانسی و حذف فرکانس های بالا و پایین آن، سیگنال دیجیتال از یک فیلتر میان گذر عبور داده می شود. این فیلتر متشکل از یک فیلتر پایین گذر و یک فیلتر بالاگذر متوالی است. هر دوی این فیلترها به صورت نرم افزاری پیاده سازی شده اند. تابع تبدیل فیلتر پایین گذر را در معادله ی ۱-۳ مشاهده می کنیم.

$$H(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2} \quad (1-3)$$

معادله ی تفاضلی این فیلتر به صورت معادله ی ۲-۳ در خواهد آمد.

^۱ADC

^۲Sampling rate

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T) \quad (۲-۳)$$

فرکانس قطع این فیلتر پایین‌گذر ۱۱ هرتز و بهره‌ی آن ۳۶ است. یک فیلتر بالاگذر به صورت سری با این فیلتر قرار می‌گیرد که تابع تبدیل آن به صورت معادله‌ی ۳-۳ است.

$$H(z) = \frac{(-1 + 32z^{-16} + z^{-32})}{(1 + z^{-1})} \quad (۳-۳)$$

که معادله‌ی تفاضلی آن به صورت معادله‌ی ۴-۳ خواهد بود.

$$y(nT) = 32x(nT - 16T) - [y(nT - T) + x(nT) - x(nT - 32T)] \quad (۴-۳)$$

این فیلتر فرکانس‌های بالای ۵ هرتز را عبور می‌دهد و بهره‌ی آن ۳۲ است. از توالی این دو فیلتر، فیلتر میان‌گذری به دست می‌آید که فرکانس‌های ۵ تا ۱۱ هرتز را عبور می‌دهد که به هدف ما برای کاهش نویز نزدیک است.

۲-۱-۲-۳ مشتق‌گیر

پس از اعمال فیلترها، عمل مشتق‌گیری بر روی سیگنال انجام می‌شود. مشتق‌گیری از سیگنال، اطلاعاتی در مورد شیب آن در بازه‌ی QRS فراهم می‌کند. تابع انتقال این فیلتر به صورت معادله‌ی ۵-۳ است و معادله‌ی تفاضلی آن به صورت رابطه‌ی ۶-۳ می‌آید.

$$H(z) = \frac{(-z^{-2} - 2z^{-1} + 2z + z^2)}{8T} \quad (۵-۳)$$

$$y(nT) = \frac{-x(nT - 2T) - 2x(nT - T) + 2x(nT + T) + x(nT + 2T)}{8T} \quad (۶-۳)$$

۳-۱-۲-۳ مجذورکننده

پس از مشتق گیری، مجذور سیگنال به صورت نقطه به نقطه به دست می آید. معادله‌ی تفاضلی فیلتر در این بخش به صورت معادله‌ی ۷-۳ است. اعمال این فیلتر بر روی خروجی مشتق گیر، باعث می شود تمامی نقاط سیگنال مثبت شده و به دلیل انجام عمل مربع کردن، فواصل نقاط گسسته‌ی سیگنال تشدید شود.

$$y(nT) = [x(nT)]^2 \quad (۷-۳)$$

۴-۱-۲-۳ انتگرال گیر با پنجره‌ی لغزان

در این مرحله سیگنال مربع شده وارد یک انتگرال گیر می شود. هدف از این کار، به دست آوردن اطلاعاتی در مورد شکل موج سیگنال، علاوه بر اطلاعات مربوط به شیب موج R است که در مراحل قبل به دست آمد. معادله‌ی تفاضلی این انتگرال گیر به صورت معادله‌ی ۸-۳ است.

$$y(nT) = \frac{x(nT - (N - 1)T) + x(nT - (N - 2)T) + \dots + x(nT)}{N} \quad (۸-۳)$$

که در آن N تعداد نمونه‌ها در طول پنجره‌ی انتگرال گیر است. N به صورت تجربی به دست می آید و در تشخیص نهایی R اهمیت زیادی دارد. به طور معمول N باید تقریباً به اندازه‌ی عریض ترین بازه‌ی QRS باشد. در صورتی که پنجره بیش از حد عریض باشد، در هنگام انتگرال گیری، شکل موج QRS با موج T ترکیب می شود. اگر پنجره بیش از حد کوتاه باشد، کل بازه‌ی QRS را در بر نمی گیرد و در این بازه تعداد زیادی قله تولید خواهد شد. این مقدار به طور تجربی به دست

آمده و با نرخ نمونه‌برداری ارتباط دارد. در این پروژه طول پنجره ۷۰ در نظر گرفته شده است.

۳-۱-۵ تعیین موقعیت قله‌های R با کمک مقدارهای آستانه

موج QRS هم‌زمان با لبه‌ی بالارونده‌ی انتگرال‌گیر رخ می‌دهد، و طول بازه‌ی این لبه برابر با طول بازه‌ی QRS است. به این ترتیب، می‌توان موقعیت زمانی QRS را از روی جایگاه لبه‌ی بالارونده تعیین کرد. با استفاده از این اطلاعات، و همین‌طور اطلاعات مربوط به شیب منحنی QRS در این بازه، می‌توان نقطه‌ی ثابتی را به عنوان موقعیت قله‌ی R به دست آورد. برای تعیین درست موقعیت قله‌ی R تعدادی ولتاژ آستانه^۳ اعمال می‌شوند و به نسبت بالاتر یا پایین‌تر بودن ولتاژ هر نمونه از آن‌ها، وجود یا عدم وجود قله تشخیص داده می‌شود. این آستانه‌ها با گذشت زمان با نویز تطبیق می‌یابند. در مجموع دو سری ولتاژ آستانه داریم که هر کدام شامل دو آستانه هستند. در هر یک از این دو سری، آستانه‌ی بالاتر برای تحلیل اولیه‌ی سیگنال استفاده می‌شود، و در صورتی که در یک بازه‌ی زمانی مشخص QRS ای تشخیص داده نشده باشد، لازم است در این بازه از تکنیک جستجوی برگشتی^۴ استفاده شود. در این تکنیک در این بازه‌ی زمانی از آستانه‌های پایین‌تر برای تشخیص QRS استفاده می‌شود. روابط این آستانه‌ها در معادله‌ی ۳-۹ مشاهده می‌شوند. در این روابط، PEAK1 بالاترین ولتاژ سیگنال به طور کلی، SPKI تخمین جاری از بالاترین ولتاژ سیگنال و NPKI تخمین جاری از بالاترین ولتاژ نویز در هر لحظه است. همچنین TH I1 اولین مقدار آستانه‌ی اعمال‌شده بر روی سیگنال انتگرال‌گیری‌شده و TH I2 دومین مقدار آستانه و نصف مقدار آستانه‌ی اول است.

$$SPKI = 0.125PEAKI + 0.875SPKI$$

$$NPKI = 0.125PEAKI + 0.875NPKI$$

(۳-۹)

$$TH I1 = NPKI + 0.25(SPKI - NPKI)$$

$$TH I2 = 0.5THI1$$

^۳Threshold

^۴Search-back

برای این که یک نمونه به عنوان قله‌ی R تشخیص داده شود، باید مقداری بالاتر از TH I1 داشته باشد. در صورتی که یک قله‌ی R در فرایند جستجوی برگشتی تشخیص داده شود، مقدار SPKI به صورت رابطه‌ی ۳-۱۰ به‌روز خواهد شد.

$$SPKI = 0.25PEAKI + 0.75SPKI \quad (۳-۱۰)$$

۳-۲-۲ پیاده‌سازی الگوریتم تشخیص QRS بر روی بستر سخت‌افزاری

ورودی این بخش، سیگنال دیجیتال دریافت شده از سنسور ضربان قلب است. نحوه‌ی تولید این سیگنال و نوع سنسور به‌کاررفته برای آن کاملاً به کاربرد بستگی داشته و در این پروژه تأکیدی بر روی آن نیست. محاسبات انجام‌شده در الگوریتم تشخیص QRS، به مقدار نرخ نمونه‌برداری سیگنال ضربان قلب وابسته است. پارامترهای الگوریتم پیاده‌سازی شده در این بخش، برای نرخ نمونه‌برداری ۳۶۰ نمونه بر ثانیه بهینه شده‌اند و از این روی، لازم است نرخ نمونه‌برداری سیگنال دیجیتال ورودی، مساوی با ۳۶۰ یا نزدیک به آن باشد.

خروجی این بخش، موقعیت زمانی قله‌ی R در هر یک از بازه‌های QRS تشخیص‌داده‌شده در ضربان قلب است. به بیان دیگر، الگوریتم برخی از نمونه‌ها در سیگنال را به عنوان قله‌ی R تشخیص داده و شماره‌ی آن نمونه را به عنوان خروجی برمی‌گرداند. این مقادیر باید برای انجام پردازش‌های آینده به سرور ارسال شوند. از آن‌جا که از کل سیستم انتظار بی‌درنگ‌بودن داریم، علاوه بر تشخیص بی‌درنگ QRS لازم است دریافت داده‌های خام از حسگر و همین‌طور فرستادن قله‌های R تشخیص‌داده‌شده به سرور نیز به صورت بی‌درنگ و در حین تشخیص QRS انجام شود. به بیان بهتر، در چنین کاربردی انجام تشخیص QRS بر روی ضربان قلب به طور کامل و سپس فرستادن تمامی Rهای تشخیص‌داده‌شده به سرور قابل قبول نخواهد بود. کارهای انجام‌شده در این بخش را می‌توان در قالب موارد زیر بیان کرد.

۳-۲-۲-۱ دریافت داده‌های خام جدید از حسگر

در این بخش، هدف بر این است که رفتار یک حسگر دیجیتال ضربان قلب با نرخ نمونه‌برداری ۳۶۰ نمونه بر ثانیه شبیه‌سازی شود. بهترین راه‌حل برای این کار، استفاده از ارتباط سریال بین مازول و یک رایانه (به جای حسگر) تشخیص داده شد. با فرض این که داده‌های چنین حسگری قبلاً دریافت و بر روی رایانه ذخیره شده باشد، در صورتی که در هر ثانیه ۳۶۰ نمونه از رایانه به ESP ارسال کنیم، رفتار یک حسگر دیجیتال با نرخ نمونه‌برداری ۳۶۰ را شبیه‌سازی کرده‌ایم.

پیاده‌سازی این بخش به این صورت انجام شد که پایه‌های RX و TX ماژول ESP8266 به پورت سریال یک کامپیوتر وصل شد و داده‌های دیجیتال ضربان قلب که قبلاً به وسیله‌ی یک حس‌گر دیجیتال تولید شده بودند، به وسیله‌ی اسکرپتی در کامپیوتر به ESP8266 ارسال شدند. در هر ثانیه ۳۶۰ مقدار از مقادیر ذخیره شده با نرخ باد ۱۱۲۵۰۰ بیت بر ثانیه به ESP8266 ارسال شدند. ESP8266 این داده‌ها دریافت کرده و پردازش‌های آینده را بر روی آن‌ها انجام خواهد داد. این ماژول به طور دائم در حال اجرای الگوریتم تشخیص QRS بر روی داده‌هایی که قبلاً دریافت کرده است می‌باشد، و در این حین داده‌های جدیدی نیز از سمت رایانه (حس‌گر) دریافت می‌کند.

۳-۲-۲-۳ اعمال الگوریتم و فرستادن شماره‌ی نمونه به سرور در صورت تشخیص قله

هدف این بخش این است که ماژول ESP8266 الگوریتم تشخیص QRS را بر روی نمونه‌هایی که دریافت می‌کند اجرا کرده و در صورت تشخیص قله، موقعیت زمانی آن را برای سرور بفرستد. در همین حین، هر لحظه نمونه‌های جدیدی از طریق ارتباط سریال دریافت می‌شوند. چالش به‌وجودآمده در این مرحله این است که این نمونه‌های جدید نباید از دست بروند. یک راه حل ممکن برای این موضوع، پیاده‌سازی نوعی مکانیزم چنددخی^۵ در ESP8266 است. در یکی از نخ‌ها، داده‌های جدید دریافت شوند و در نخ دیگر الگوریتم بر روی داده‌های موجود اجرا شود.

با بررسی‌های انجام‌شده دریافت شد که پیاده‌سازی چنددخی بر روی ESP8266 پیچیدگی بالایی داشته و کارا نمی‌باشد. به جای پیاده‌سازی این روش، از امکان ایجاد وقفه‌ی سریال در هنگام دریافت داده استفاده شد. ESP8266 امکان دریافت داده‌ها به صورت مبتنی بر وقفه را دارد، که در کتابخانه‌ی HardwareSerial به طور کامل پیاده‌سازی شده است. نحوه‌ی پیاده‌سازی به این شکل است که به محض ورود داده‌ی سریال جدید، ESP8266 کار خود را رها کرده و به وقفه سرویس می‌دهد. در روتین وقفه، کاراکتر تازه وارد از طریق ارتباط سریال، در بافر سریال ESP8266 می‌شود. سپس برنامه از روتین وقفه خارج شده و به ادامه‌ی کار خود باز می‌گردد. با استفاده از این امکان ESP8266 قادر است به طور همزمان با اجرای الگوریتم، نمونه‌های جدید را دریافت کند.

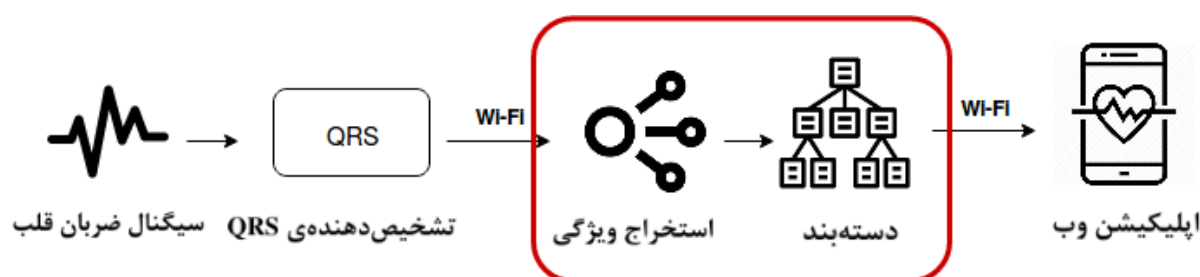
به دلیل محدود بودن حجم بافر سریال داخلی موجود در ESP8266، نیاز به پیاده‌سازی یک مکانیزم بافرینگ در خود کد نیز وجود دارد. برای جلوگیری از سرریز کردن بافر سریال، در ابتدای هر لوپ اجرای برنامه‌ی ESP8266 به این بافر سرکشی شده و داده‌های جدید را از آن بر می‌داریم و در بافری که خود پیاده‌سازی کرده‌ایم قرار می‌دهیم. این بافر برای

^۵Multithreading

اطمینان حجم بیشتری دارد و با استفاده از آرایه پیاده‌سازی شده‌است. داده‌های جدید در این آرایه می‌مانند، تا وقتی که نوبت پردازش و انجام الگوریتم روی آن‌ها فرا برسد.

۳-۳ عملیات پردازش سمت سرور

بخش دوم سیستم با استفاده از الگوریتم‌های یادگیری ماشین، و بر روی یک سرور پیاده‌سازی شده‌است. نتایج پیش‌پردازش‌های انجام شده در بخش قبل در سرور دریافت شده و ویژگی‌های هر ضربان استخراج می‌شود. سپس با استفاده از این ویژگی‌ها، عمل دسته‌بندی ضربان‌ها انجام شده و به هر ضربان یک برچسب که تعیین‌کننده‌ی کلاس ضربان است، اختصاص داده می‌شود. در شکل ۲-۳ مراحل پیاده‌سازی شده در این بخش در کادر مشخص شده‌اند. سرور مشخص شده در این تصویر، یک دامنه است که کاربران می‌توانند با وارد کردن آدرس آن در مرورگرهای خود، از هر کجا به آن متصل شوند.



شکل ۲-۳ بخش پردازش سمت سرور در مراحل تشخیص آریتمی

۱-۳-۳ نحوه‌ی اجرای الگوریتم یادگیری

۱-۱-۳-۳ استخراج ویژگی‌ها

در فرایند دسته‌بندی ضربان‌ها، یک مدل SVM با استفاده از ویژگی‌های استخراج شده ساخته می‌شود. همان‌طور که اشاره شد، بازه‌های RR و میزان نوسانات آن‌ها ظرفیت بالایی در تشخیص آریتمی دارند، از طرف دیگر، در صورت استفاده از این ویژگی، داده‌های ارسالی از سمت سخت‌افزار متصل به بیمار، حجم پایینی خواهند داشت و در نتیجه تاخیر ارسال داده‌ها به سرور به حداقل خواهد رسید، که برای کاربرد بی‌درنگ ما مناسب‌تر است به همین دلیل از بازه‌های RR برای تولید ویژگی‌ها استفاده می‌شود. برای تولید ویژگی‌ها، با دریافت هر قله‌ی R جدید در سرور، مقادیر زیر محاسبه می‌شوند:

• Pre-RR: این مقدار فاصله‌ی بین ضربان جاری (مربوط به قله‌ی R دریافت‌شده و قله‌ی ضربان قبلی را نشان می‌دهد.

• Post-RR: این مقدار فاصله‌ی زمانی بین قله‌ی R تازه دریافت‌شده و قله‌ی R بعدی را نشان می‌دهد.

• Local-RR: میانگین ۱۰ مقدار Pre-RR گذشته در این مقدار محاسبه می‌شود.

• Global-RR: میانگین مقادیر Pre-RR تولیدشده در ۱۰ دقیقه‌ی گذشته در این مقدار محاسبه می‌شود.

در سرور میانگین هر یک از این چهار مقدار محاسبه شده و با رسیدن مقادیر جدید، به‌روز می‌شود. از این میانگین سپس برای نرمال‌سازی ویژگی‌هایی که تا کنون استخراج شده‌اند استفاده می‌شود، به این صورت که هر یک از این چهار ویژگی به میانگین خود تقسیم شده و چهار ویژگی دیگر را می‌سازند. در انتها ۸ ویژگی از بازه‌های RR تولید شده و به مدل دسته‌بند تحویل داده می‌شوند.

۳-۱-۳ ساخت مدل SVM

پردازش‌های سمت سرور و الگوریتم SVM به کمک زبان پایتون و فریم‌ورک Django پیاده‌سازی شدند. زبان پایتون به دلیل پشتیبانی از کتابخانه‌های متنوعی برای پیاده‌سازی انواع الگوریتم‌های یادگیری، برای استفاده در این کاربرد مناسب تشخیص داده شد. برای پیاده‌سازی الگوریتم SVM در این زبان، از کتابخانه‌ی scikit-learn استفاده کردیم. برای ساخت و آموزش یک مدل SVM، نمونه‌ای از کلاس SVC (از کلاس‌های عضو کتابخانه‌ی scikit-learn که برای دسته‌بندی‌های چنددسته‌ای استفاده می‌شود) ساخته شده و ویژگی‌های استخراج‌شده از داده‌های آموزش، به علاوه‌ی پارامترهای ضروری برای مدل SVM به آن داده شدند.

تابع کرنل در این مدل، RBF در نظر گرفته شده و پارامترهای مربوط به آن تعیین شده‌است. مقدار گاما در این

مدل‌ها برابر با مقدار پیش‌فرض گاما در مدل‌های SVM قرار داده شد که مساوی با $\frac{1}{\text{number of features}}$ است.

برای تعیین پارامتر C در این مدل، از روش اعتبارسنجی متقابل ۱۰ لایه‌ای استفاده کردیم. در این روش، پارامتر C از مقدار ۰/۰۰۱ تا ۱۰۰۰۰ به صورت لگاریتمی تغییر داده شده و به ازای هر مقدار، معیارهای کارایی محاسبه شدند. اعتبارسنجی متقابل در انتها مقدار ۰/۰۰۱ را به عنوان بهترین مقدار در این بازه برای پارامتر C به دست آورد.

سپس مدل ساخته شده و آموزش داده شد. در مرحله‌ی بعد، مدل ساخته شده با کمک توابع کلاس SVC بر روی

مجموعه داده‌های تست امتحان شدند، و از استراتژی رای‌دهی OVO^۶ برای این کار استفاده شد که در بخش بعد به تفصیل توضیح داده خواهد شد.

۳-۱-۳-۳ استراتژی رای‌دهی

مدل‌هایی که با استفاده از SVM ساخته می‌شوند، تنها برای مسائل دسته‌بندی دودویی^۷ قابل استفاده هستند. در این مسائل، هدف جداسازی دو کلاس از داده‌ها است. این محدودیت باعث شده است راه‌حل‌هایی ارائه شود تا بتوان از SVM در مسائل دسته‌بندی چندکلاسی نیز استفاده کرد. دو راه‌حل اصلی و پرکاربرد برای حل این گونه مسائل، روش OVO و روش OVR^۸ است. در هر دوی این روش‌ها، مدل SVM برای حل مسئله‌ای دودویی آموزش داده می‌شود و سپس با استراتژی‌هایی، این دسته‌بندی به چند کلاس تعمیم داده می‌شود.

در روش OVR برای یک مسئله دسته‌بندی k کلاسی، به ازای هر کلاس، یک مدل SVM آموزش داده می‌شود و تعیین می‌کند آیا یک نمونه به این کلاس تعلق دارد یا به بقیه‌ی کلاس‌ها. در این روش k دسته‌بند دودویی داریم که بر روی نمونه‌ها اعمال می‌شوند و در انتهابه ازای هر نمونه، k نتیجه به دست می‌آید. در روش OVO تعداد $\frac{k(k-1)}{2}$ مدل SVM دودویی متمایز آموزش داده می‌شود. این تعداد برابر با ترکیب ۲ از k کلاس، و به بیان دیگر، تعداد روش‌های انتخاب دو کلاس از بین این k کلاس است. هر یک از این مدل‌ها یک جفت از کلاس‌های هدف را دریافت کرده و تشخیص می‌دهد یک نمونه به کدام یک از این دو کلاس تعلق دارد. در این روش به ازای هر نمونه، $C(n, 2)$ نتیجه به دست می‌آید. در هر دوی این روش‌ها، برای تعیین نتیجه‌ی نهایی دسته‌بندی هر نمونه، یک استراتژی رای‌دهی^۹ مورد نیاز است. روش OVO نتایج بهتری در مواردی که داده‌ها نامتعادل هستند از خود نشان داده‌است. همچنین این روش در مواردی که تعداد نمونه‌ها بسیار زیاد است، زمان کمتری برای یادگیری نیاز دارد. [۱] به همین دلیل در این پروژه از OVO استفاده شده‌است.

با توجه به این که در این کار، ۴ کلاس هدف داریم، در مرحله‌ی یادگیری، ۶ مدل دودویی ساخته شده و بر روی هر نمونه اعمال می‌شود. در واقع در این جا ۶ ابرصفحه ساخته می‌شود که هر کدام، نمونه‌های یک جفت از کلاس‌های هدف را از هم تمایز می‌دهد. در انتهای این مرحله، ماتریس تابع تصمیم‌گیری^{۱۰} تولید می‌شود که طول آن برابر با تعداد نمونه‌های

^۶One Versus One

^۷Binary classification

^۸One versus Rest

^۹Voting strategy

^{۱۰}Decision function

مجموعه‌ی آموزش و عرض آن برابر با ۶ است. بر روی هر یک از این ۶ مقدار تعیین شده برای هر نمونه، لازم است یک استراتژی رای‌دهی اعمال شود. این استراتژی به صورتی که در ادامه توضیح داده خواهد شد پیاده‌سازی شده است. هر یک از مقادیر تابع تصمیم‌گیری نتیجه‌ی دسته‌بندی بین یک جفت از کلاس‌ها را نشان می‌دهد. این نتیجه را مثبت یا منفی بودن مقدار نشان می‌دهد. با توجه به این مقادیر، کلاسی که بیشترین مقدار مثبت را برای یک نمونه دریافت کرده‌باشد، به عنوان کلاس برگزیده برای آن نمونه انتخاب می‌شود.

۳-۳ نحوه‌ی پردازش داده‌های دریافت‌شده در سرور

مراحل‌ی که پیش‌تر توضیح داده شدند، به خصوص اعتبارسنجی متقابل، از نظر محاسبات و زمان اجرا بسیار هزینه‌بر هستند. این بخش‌ها به صورت آفلاین و بر روی کامپیوتر شخصی اجرا شدند. خروجی این مراحل یک مدل SVM است که توانایی پیش‌بینی برچسب داده‌های ناشناخته و جدید را دارد. کتابخانه‌ی scikit-learn این امکان را می‌دهد که مدل SVM حساب‌شده، در یک فایل باینری ذخیره شود. هر بار که نیاز به پیش‌بینی برچسب داده‌ی جدیدی بود، مدل SVM را می‌توان از روی این فایل بارگیری کرده و با پاس دادن ویژگی‌های داده‌های جدید به آن، برچسب‌های پیش‌بینی‌شده را دریافت نمود. هزینه‌ی این کار به طرز چشم‌گیری کمتر از این است که مدل هر بار از روی داده‌های آموزش ساخته شده و آموزش داده شود.

پس از انتخاب بهترین پارامترها به کمک اعتبارسنجی متقابل، ساخت و ارزیابی مدل به صورت آفلاین انجام شد. این مدل بر روی فایل باینری ذخیره شده و به سرور انتقال داده شد. برای پیاده‌سازی عملیات سمت سرور، از محیط توسعه‌ی فراهم‌شده به وسیله‌ی وبسایت pythonanywhere.com استفاده شد. این محیط، قابلیت میزبانی اپلیکیشن‌های وب توسعه‌داده‌شده با پایتون را فراهم می‌کند.

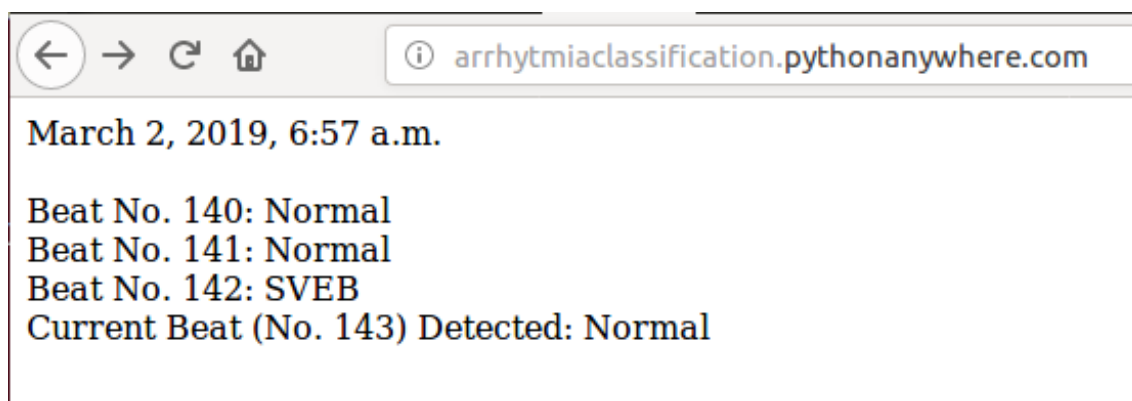
نحوه‌ی عملکرد کد سمت سرور به این صورت است که به محض دریافت داده‌های پیش‌پردازش‌شده‌ی جدید از سمت سخت‌افزار، این داده‌ها در پایگاه‌داده‌ی خود سرور ذخیره می‌شوند. سخت افزار برای فرستادن مقدار جدیدی به نام value به

سرور، هر بار یک درخواست HTTP به این شکل زیر به سرور می‌فرستد: `GET /store/{value}`

کاربر برای دیدن نتایج پردازش‌ها، کافی است URL دامنه را در مرورگر خود وارد کند، و در واقع یک درخواست HTTP GET به مسیر / بزند. این درخواست در سرور به این صورت مدیریت می‌شود که ابتدا مدل SVM از روی فایل بارگیری شده، سپس آخرین داده‌های پیش‌پردازش‌شده که تا کنون در پایگاه‌داده ذخیره شده‌اند، به مدل داده می‌شوند تا

کلاس آن‌ها پیش‌بینی شود. سپس نتیجه در مرورگر به کاربر نشان داده خواهد شد.

برای آسان‌تر بودن استفاده برای کاربر، صفحه‌ی وب طراحی شده به صورت خودکار و هر یک ثانیه یک بار بارگیری می‌شود. با فرض این که سخت‌افزار به طور مرتب قله‌های R تشخیص داده‌شده را برای سرور بفرستد، در هر ثانیه یک درخواست GET به مسیر / زده شده و کلاس‌های داده‌های جدید به وسیله‌ی مدل، پیش‌بینی و بر روی صفحه‌ی وب نمایش داده می‌شوند. با توجه به این که به طور تقریبی در هر ثانیه یک ضربان قلب تولید می‌شود، چنین کاربردی به صورت بی‌درنگ و با تاخیری اندک نتایج دسته‌بندی هر ضربان را به کاربر نشان می‌دهد. نمونه‌ای از خروجی نشان داده‌شده به کاربر در مرورگر، در شکل ۳-۳ قابل مشاهده است.



شکل ۳-۳ خروجی نشان داده‌شده در مرورگر

۳-۳-۳ داده‌های مورد بررسی در الگوریتم یادگیری

به منظور استانداردسازی الگوریتم‌های گوناگون تشخیص آریتمی، لازم است ارزیابی این الگوریتم‌ها بر روی مجموعه داده‌های استاندارد و مشترکی صورت بگیرد تا در مقایسه‌ی نتایج حاصل از آن‌ها با دقت کافی حاصل شود. برای این منظور، تعدادی پایگاه داده از نمونه‌های ضربان قلب توسط موسسه‌های گوناگون گردآوری شده است. انجمن پیشبرد ابزار دقیق پزشکی^{۱۱} که به اختصار AAMI نامیده می‌شود، تعدادی از این پایگاه داده‌ها را به عنوان منابع استاندارد داده برای ارزیابی الگوریتم‌های تشخیص آریتمی معرفی کرده و هم‌چنین قراردادهایی برای اجرای عملیات ارزیابی الگوریتم‌ها در آن تعیین کرده است، تا کاملاً از تکرارپذیری و قابل مقایسه بودن نتایج آزمایش‌های متفاوت اطمینان حاصل شود. در این استاندارد،

^{۱۱} Association for the Advancement of Medical Instrumentation (AAMI)

استفاده از ۵ پایگاه داده‌ی زیر توصیه شده است:

- پایگاه داده‌ی MIT-BIH^{۱۲} شامل ۴۸ نوار قلب ۳۰ دقیقه‌ای
- پایگاه داده‌ی EDB: شامل ۹۰ نوار قلب ۲ ساعته
- پایگاه داده‌ی AHA شامل ۸۰ نوار قلب ۳۵ دقیقه‌ای
- پایگاه داده‌ی CU شامل ۳۵ نوار قلب ۸ دقیقه‌ای
- پایگاه داده‌ی NST شامل ۱۲ نوار قلب ۳۰ دقیقه‌ای

از بین این موارد، MIT-BIT که در این پروژه از آن استفاده کرده‌ایم، اولین پایگاه داده‌ی به وجود آمده برای این منظور، و پرکاربردترین مجموعه داده برای دسته‌بندی و ارزیابی الگوریتم‌های تشخیص آریتمی است. [۲۰] در ادامه این پایگاه داده را دقیق‌تر بررسی خواهیم کرد.

۱-۳-۳ پایگاه داده‌ی MIT-BIH

برچسب‌گذاری‌های ضربان‌ها در MIT-BIH در طی سال‌ها به طور دائم بهبود داده شده‌اند. به دلیل گستردگی داده‌ها و وجود انواع ضربان قلب در این نمونه‌ها، بیشترین پژوهش‌ها بر روی این پایگاه داده انجام گرفته‌اند. [۲۰] در MIT-BIH تمامی ضربان‌ها به وسیله‌ی یک الگوریتم تشخیص QRS از یکدیگر متمایز شده‌اند و به هر تک‌ضربان قلب، برچسبی اختصاص داده شده است که نوع آن ضربان را تعیین می‌کند. این برچسب‌ها در برای پیاده‌سازی و ارزیابی الگوریتم‌های تشخیص آریتمی ضروری هستند. نحوه‌ی انجام این برچسب‌گذاری نیز در استاندارد AAMI تعیین شده‌است. با وجود تنوع انواع ضربان قلب‌های دارای آریتمی، ترجیح AAMI بر استفاده از ۱۵ کلاس از بین این انواع است. این ۱۵ کلاس، خود به ۵ کلاس کلی‌تر طبقه‌بندی شده‌اند:

۱- ضربان‌های عادی^{۱۳} (N)

۲- ضربان‌های ناب‌جای فوق بطنی^{۱۴} (SVEB)

^{۱۲}Massachusetts Institute of Technology - Beth Israel Hospital

^{۱۳}Normal

^{۱۴}Supraventricular ectopic beats

۳- ضربان‌های نابه‌جای بطنی^{۱۵} (VEB)

۴- ضربان‌های ادغام‌شده^{۱۶} (F)

۵- ضربان‌های ناشناخته^{۱۷} (Q)

این پایگاه‌داده شامل ۴۸ نوارقلب با نرخ نمونه‌برداری ۳۶۰ هرتز است. این نوارقلب‌ها از ۴۷ بیمار گرفته شده‌اند و هر کدام مدت‌زمانی برابر با ۳۰ دقیقه دارد. هر نوارقلب، شامل نمونه‌های دو لید مجزا است. در بیشتر نوارقلب‌ها، لید اصلی که لید A نام دارد، نمونه‌ی تغییر یافته‌ای از لید II است که از الکترودهای قرار گرفته بر روی سینه به دست می‌آید. لید دوم که لید B نام دارد، در بیشتر نوارقلب‌ها لید V1 و در دیگران V2، V4 و یا V5 است. عموماً برای تشخیص آریتمی از لید اول (A) استفاده می‌شود، چرا که در این لید، موج QRS واضح‌تر است. [۲۰]

پایگاه‌داده‌های موجود از نظر تعداد ضربان‌های متعلق به هر کلاس آریتمی، شدیداً نامتعادل^{۱۸} هستند. MIT-BIH تنها پایگاه‌داده‌ای است که هر ۵ کلاس آریتمی ذکر شده را پوشش می‌دهد. اما در این پایگاه‌داده نیز، حدود ۹۰ درصد ضربان‌ها در کلاس N جای می‌گیرند و از ۱۰ درصد باقی‌مانده، حدود ۳٪، ۶٪ و ۱٪ به ترتیب متعلق به کلاس‌های SVEB، VEB و F هستند، و درصد ضربان‌های کلاس Q پایین‌تر از ۱ درصد است. [۱] به همین دلیل، لازم است در الگوریتم‌های دسته‌بندی و روش‌های ارزیابی آن‌ها، نامتعادل بودن پایگاه‌داده مد نظر قرار گیرد.

۳-۳-۲ نحوه‌ی تقسیم داده‌ها به دو مجموعه‌ی آموزش و تست

دو الگوی اصلی برای ارزیابی روش‌های اتوماتیک تشخیص آریتمی استفاده می‌شود: الگوی درون‌بیماری^{۱۹} و الگوی بین‌بیماری^{۲۰}. در الگوی اول، هیچ‌گونه محدودیتی در تقسیم پایگاه‌داده به دو بخش آموزش و تست وجود ندارد و هر یک از ضربان‌های موجود را می‌توان صرف نظر از این که متعلق به کدام بیمار است، در هر یک از این دو مجموعه جای داد. این نوع تقسیم بندی، یک نقص اساسی در این روش تقسیم‌بندی را موجب می‌شود. از آن جا که در حین یادگیری، امکان دارد مدل تولید شده بتواند الگوهای موجود در ضربان‌های یک بیمار خاص را نیز تشخیص داده و یادبگیرد، نتایج ارزیابی

^{۱۵}Ventricular ectopic beats

^{۱۶}Fusion beats

^{۱۷}Unknown beats

^{۱۸}Imbalanced

^{۱۹}Intra-patient paradigm

^{۲۰}Inter-patient paradigm

به دست آمده از الگوریتمی که با الگوی درون بیماری کار می کند، نمی تواند کاملاً قابل اعتماد باشد. چرا که به طور مطلوب، یک الگوریتم دسته بندی آریتمی باید بتواند برای هر بیماری، با دقتی معین عمل کند، حتی اگر سیستم از پیش اطلاعی در مورد آن بیمار نداشته باشد. در راستای ارزیابی واقع گرایانه تر، الگوی بین بیماری توسط Chazal و همکاران معرفی شد. [۱۲] در این الگو تقسیم بندی پایگاه داده به دو مجموعه یادگیری و ارزیابی، طوری صورت می گیرد که هیچ ضربانی از یک بیمار خاص در هر دو مجموعه به طور هم زمان حاضر نباشد. نحوه تقسیم بندی داده ها در استاندارد ارائه شده به صورت زیر می باشد:

• مجموعه داده ی اول (DS1) شامل نوار قلب های ۱۰۱، ۱۰۶، ۱۰۸، ۱۰۹، ۱۱۲، ۱۱۴، ۱۱۵، ۱۱۶، ۱۱۸، ۱۱۹، ۱۲۲،

۱۲۴، ۲۰۱، ۲۰۳، ۲۰۵، ۲۰۷، ۲۰۸، ۲۰۹، ۲۱۵، ۲۲۰، ۲۲۳ و ۲۳۰

• مجموعه داده ی دوم (DS2) شامل نوار قلب های ۱۰۰، ۱۰۳، ۱۰۵، ۱۱، ۱۱۳، ۱۱۷، ۱۲۱، ۱۲۳، ۲۰۰، ۲۰۲، ۲۱۰،

۲۱۲، ۲۱۳، ۲۱۴، ۲۱۹، ۲۲۱، ۲۲۲، ۲۲۸، ۲۳۱، ۲۳۲، ۲۳۳ و ۲۳۴

همان طور که اشاره شد، ارزیابی مدل ها با استفاده از این الگو، نتایج قابل اعتمادتری ارائه می کنند. این روش تقسیم بندی پس از معرفی، به طور گسترده ای در کارهایی که با الگوی بین بیماری کار می کنند به کار رفته است. در این پروژه نیز از این الگو برای ارزیابی بهره گرفته شده است.

۳-۴ ارزیابی نتایج حاصل از یادگیری

معیارهای دقت، صحت، صحت کلی و حساسیت (که در بخش ۲-۵ توضیح داده شدند) توسط AAMI برای ارزیابی روش های مختلف پیشنهاد شده اند. به دلیل نامتعادل بودن پایگاه داده های موجود، صحت کلی نمی تواند معیار مناسبی برای سنجش باشد. در MIT-BIH به دلیل پر تعداد بودن ضربان های متعلق به کلاس N که حدود ۹۰ درصد داده ها را به خود اختصاص داده اند، این کلاس در محاسبه ی صحت کلی بر دیگر کلاس ها غالب می شود. به همین دلیل، این معیار به طور معمول برای مقایسه ی الگوریتم ها مورد استفاده قرار نمی گیرد.

۳-۴-۱ معیارهای کارایی

علاوه بر چهار معیار ذکر شده، معیارهای دیگری توسط AAMI به عنوان معیارهای استاندارد معرفی شده اند. این معیارها از ماتریس درهم ریختگی استخراج می شوند و طبق استاندارد، استثنائاتی در محاسبه ی آن ها در نظر گرفته می شود.

برای مثال، در صورتی که الگوریتم دسته‌بندی، ضربانی از کلاس F را به اشتباه متعلق به کلاس VEB تشخیص دهد، در میزان موفقیت یا خطای الگوریتم تاثیری نخواهد داشت. همان‌طور که ذکر شد، صحت کلی نمی‌تواند میزان موفقیت یک الگوریتم در دسته‌بندی ضربان‌ها را به خوبی نشان دهد. برای غلبه بر این مشکل، یک مقدار جدید به نام اندیس $j\kappa$ توسط Mar و همکاران معرفی شده است. [۲۷] این مقدار مجموع وزن‌داری از اندیس j و کاپای کوهن (κ) است.

اندیس j (رابطه‌ی ۱۱-۳) به پیروی از استاندارد AAMI، دقت تمایزدهی مهم‌ترین کلاس‌های آریتمی، یعنی VEB و SVEB را اندازه‌گیری می‌کند. [۱]

$$j\text{ index} = Se_{SVEB} + Se_{VEB} + P_{SVEB}^+ + P_{VEB}^+ \quad (11-3)$$

کاپای کوهن مقداری قراردادی برای ارزیابی نتایج موجود در ماتریس درهم‌ریختگی است. کاپا به عنوان معیاری مقاوم‌تر به نسبت صحت کلی برای پایگاه‌داده‌های نامتعادل گزارش شده‌است. در این رابطه، P_o که احتمال مشاهده‌شده^{۲۱} نام دارد، مساوی با صحت کلی است، و مقداری دیگر به نام P_e نیز تعریف می‌شود. در محاسبه‌ی این مقدار، تعداد نمونه‌های موجود در هر کلاس لحاظ شده‌است و به این دلیل برای مقایسه‌ی عملکرد در پایگاه‌داده‌های نامتعادل مناسب است.

$$\begin{aligned} \kappa &= \frac{P_o - P_e}{1 - P_e} \\ P_o &= \frac{Nn + Ss + Vv + Ff}{\Sigma} \\ P_e &= \frac{\Sigma N \Sigma n + \Sigma S \Sigma s + \Sigma V \Sigma v + \Sigma F \Sigma f}{\Sigma^2} \end{aligned} \quad (12-3)$$

اندیس $j\kappa$ به صورت رابطه‌ی ۱۳-۳ محاسبه می‌شود. وجود κ باعث می‌شود نرخ دسته‌بندی نادرست^{۲۲} و همین‌طور تعداد نمونه‌های موجود در هر کلاس در محاسبه‌ی این معیار مد نظر قرار داده شود. به طور هم‌زمان، میزان تشخیص VEB و SVEB نیز که در محاسبه‌ی اندیس j لحاظ شده‌اند، در این مقدار دخالت دارند. تمامی این مقادیر، اندیس $j\kappa$ را به

^{۲۱} Observed probability

^{۲۲} Misclassification rate

معیاری مناسب برای مقایسه‌ی الگوریتم‌های دسته‌بندی آریتمی تبدیل کرده است.

$$j\kappa index = w_1\kappa + w_2(j index) \quad (۱۳-۳)$$

فصل ۴

نتایج به دست آمده

در این کار، آزمایش‌هایی با هدف بهینه کردن زمان پاسخ کل الگوریتم و دقت الگوریتم دسته‌بندی انجام شد. همچنین برای ارتباط بهتر کاربران (بیمار و پزشک) با سیستم، یک اپلیکیشن تحت وب برای دسترسی کاربر به نتایج دسته‌بندی طراحی و پیاده‌سازی شد. در ادامه به نتایج به دست آمده در این بخش‌ها می‌پردازیم.

۱-۴ زمان پاسخ سیستم

در این کاربرد، زمان پاسخ را مدت‌زمان بین تولید یک ضربان قلب در سیگنال نوار قلب، تا لحظه‌ای که کلاس آن ضربان به کاربر نشان داده می‌شود در نظر گرفته‌ایم. این زمان معیاری برای بررسی سرعت و میزان کارآمد بودن سیستم، به عنوان یک سیستم بی‌درنگ بوده و از این جهت اهمیت بالایی دارد.

همان‌طور که پیش‌تر توضیح داده شد، در ابتدا سخت‌افزار با دریافت یک ضربان قلب، الگوریتم تشخیص QRS را بر روی آن اجرا کرده و قله‌ی R را در ضربان تشخیص می‌دهد. مدت‌زمانی که طول می‌کشد تا این عمل انجام شود را t_{QRS} نامیده‌ایم. این زمان به طور متوسط ۳۲۰ میکروثانیه محاسبه شد که به دلیل ناچیز بودن در برابر زمان‌های محاسبه‌شده‌ی دیگر، در محاسبات لحاظ نشد. سخت‌افزار پس از تشخیص یک قله‌ی R آن را بلافاصله برای سرور می‌فرستد. این مدت‌زمان، زمان ارسال به سرور (t_{send}) نامیده شده و به طور متوسط برابر با ۰/۶ ثانیه محاسبه شده است.

قله‌ی R تشخیص‌داده‌شده به محض رسیدن به سرور، در پایگاه‌داده ذخیره خواهد شد. مدل زمان انجام این عمل که t_{store} نام دارد در بدترین حالت ۳۰ میلی‌ثانیه به دست آمد. برای دیدن نتیجه‌ی هر ضربان، لازم است صفحه دوباره بارگذاری شده و عملیات پیش‌بینی انجام شود. صفحه با نرخ یک بار در ثانیه بارگذاری می‌شود و در بدترین حالت، یک ثانیه بعد درخواستی برای پیش‌بینی برچسب ضربانی که هم اکنون در سرور ذخیره شده‌است، از طرف مرورگر به سرور داده خواهد شد. این زمان را $t_{refresh}$ می‌نامیم.

پس از ارسال درخواست، سرور مدتی را صرف پردازش داده‌ی جدید و نمایش نتیجه در صفحه‌ی وب می‌کند. این مدت زمان که $t_{predict}$ نامیده شد نیز در بیشترین حالت ۵۰ میلی‌ثانیه به دست آمد. به این ترتیب می‌توان کل مدت‌زمان پاسخ را طبق معادله‌ی ۱-۴ محاسبه کرد.

جدول ۱-۴ نتایج دسته‌بندی در کلاس‌های مختلف ضربان و به صورت میانگین

Beat Class	Sensitivity	Precision	Accuracy
N	0.7657	0.9865	0.7831
SVEB	0.4717	0.2653	0.9243
VEB	0.7854	0.4773	0.9299
F	0.9201	0.0572	0.8809
Mean	0.7357	0.4466	0.8796

$$\begin{aligned}
 t_{\text{response}} &= t_{\text{send}} + t_{\text{store}} + t_{\text{refresh}} + t_{\text{predict}} \\
 &= 600ms + 30ms + 1s + 50ms = 1/680s
 \end{aligned}
 \tag{۱-۴}$$

به این ترتیب مدت‌زمانی کم‌تر از ۲ ثانیه برای زمان پاسخ ضمانت می‌شود.

۲-۴ معیارهای کارایی نهایی الگوریتم دسته‌بندی

همان‌طور که در بخش ۱-۴-۳-۳ اشاره شد، مهم‌ترین معیاری که در این کار برای سنجش میزان موفقیت الگوریتم دسته‌بندی مورد استفاده قرار دادیم، اندیس k_c است. روش اعتبارسنجی متقابل مقدار $0/0/0/1$ را به عنوان بهترین مقدار برای پارامتر C تعیین نمود. پس از قراردادن این مقدار برای C و ساخت و ارزیابی مدل، مقدار اندیس k_c برابر با $0/428$ به دست آمد. در جدول ۱-۴ مقادیر به دست آمده برای دیگر معیارهای کارایی مشاهده می‌شوند.

همچنین مقدار k_c برابر با $0/3567$ و مقدار اندیس k_c برابر با $1/9998$ به دست آمد. همان‌طور که اشاره شد، اندیس k_c به کمک معادله‌ی ۱۳-۳ با استفاده از این مقادیر محاسبه می‌شود.

فصل ۵

نتیجه‌گیری و کارهای آینده

در این کار، یک سیستم تشخیص و دسته‌بندی آریتمی قلبی پیاده‌سازی شد. این سیستم توانایی کارکردن به صورت بی‌درنگ را دارد و در طول مدت‌زمان کم‌تر از ۲ ثانیه، کلاس آریتمی ضربان دریافت‌شده را تشخیص می‌دهد. دقت الگوریتم دسته‌بندی پیاده‌سازی‌شده به طور میانگین ۰/۴۴ به دست آمد، هم‌چنین حساسیت میانگین این دسته‌بندی ۰/۷۳ و $j\kappa$ index آن ۰/۴۳ محاسبه شد.

این سیستم به دو بخش کلی پیش‌پردازش بر روی سخت‌افزار و پردازش اصلی بر روی سرور تقسیم می‌شود. بخش اول بر روی یک ماژول ESP8266 پیاده شد که باید به یک سنسور دیجیتالی ضربان قلب متصل شود. در این بخش یک الگوریتم تشخیص QRS با روش پن-تامپکینز و با هدف تشخیص قله‌های R هر ضربان بر روی سیگنال نوار قلب اجرا شد و نتیجه‌ی این پیش‌پردازش به سرور ارسال گردید.

در بخش دوم، یک مدل دسته‌بندی SVM به صورت آفلاین آموزش داده شد و سپس مدل آموزش‌داده‌شده به صورت فایل ذخیره شده و به سرور انتقال داده شد. یک کد سمت سرور برای دریافت درخواست‌ها و پیش‌بینی کلاس ضربان‌های جدید پیاده‌سازی شد.

الگوریتم دسته‌بندی بر روی پایگاه‌داده‌ی MIT-BIH اجرا شد و برای تقسیم داده‌های این پایگاه‌داده به دو مجموعه‌ی آموزش و تست، از الگوی بین‌بیماری بهره گرفته شد. برای ارزیابی کارایی این دسته‌بندی، از معیار $j\kappa$ index استفاده شد که نمای مناسبی از میزان کارایی یک الگوریتم دسته‌بندی آریتمی ارائه می‌کند.

از مهم‌ترین نیازمندی‌های این پروژه، سرعت بالا و بی‌درنگ بودن تشخیص آریتمی است که با توجه به زمان پاسخ به‌دست‌آمده، می‌توان گفت این نیازمندی برآورده شده‌است. هم‌چنین قابل‌حمل بودن سخت‌افزار همراه بیمار از نیازمندی‌های دیگر بود که با پیاده‌سازی بخش سخت‌افزاری کار بر روی ماژول ESP8266 که حجم و مساحت کوچکی دارد، تا حد خوبی برآورده شده‌است.

نیازمندی دیگر این کار، دقت بالای الگوریتم دسته‌بندی است. در این کار در مرحله‌ی استخراج ویژگی، تنها از بازه‌های RR ضربان‌های قلب استفاده شد و ویژگی‌هایی با استفاده از این بازه‌ها استخراج شدند. دلیل این امر، کارایی بالایی بود که این ویژگی در کارهای گذشته از خود نشان داده‌است. حجم پایین ویژگی‌های استخراج شده و سبک‌تر بودن محاسبات مورد نیاز بر روی سخت‌افزار همراه بیمار نیز باعث شد ویژگی RR مناسب تشخیص داده شود، چرا که ما را به هدف بی‌درنگ بودن سیستم و قابل‌حمل بودن سخت‌افزار آن نزدیک می‌نماید.

در برخی از الگوریتم‌های دسته‌بندی موجود از ویژگی‌های دیگری چون ویژگی‌هایی که شکل موج سیگنال را مورد بررسی قرار می‌دهند، و یا ترکیب چندین ویژگی از این دست استفاده شده‌است. این ویژگی‌ها به دلیل توصیف بهتر سیگنال ضربان قلب (به نسبت ویژگی RR) بعضاً مدل‌هایی با $j\kappa$ index بالاتری از آنچه در این کار به دست آمد ارائه کرده‌اند، و در حال حاضر بالاترین $j\kappa$ index به دست آمده در کارهای گذشته ۰/۶۶۳ است [۲۸].

در کارهای آینده می‌توان از چند مدل SVM که هر یک با یک مجموعه از ویژگی‌ها آموزش داده شده‌اند و ترکیب نتایج آن‌ها، برای ساخت یک مدل SVM قوی‌تر بهره برد. در کار پیش رو، در مرحله‌ی تشخیص QRS، تنها از ولتاژ یک لید (MLII) استفاده شد که واضح‌ترین ترکیب QRS را بین لیدهای قلبی نمایش می‌دهد. در آینده می‌توان الگوریتم‌هایی طراحی نمود که از لیدهای بیشتری بهره می‌برند، و تاثیر این کار را بر بالارفتن دقت تشخیص QRS بررسی کرد.

مراجع

- [1] V. Mondéjar-Guerra, J. Novo, J. Rouco, M. G. Penedo, and M. Ortega, "Heartbeat classification fusing temporal and morphological information of ecgs via ensemble of classifiers," vol.47, pp.41–48, 2019.
- [2] "The Basics of ECG Interpretation," Jan. 2016. [Online]. Available: <https://www.medicalexamprep.co.uk/the-basics-of-ecg-interpretation-part-1-anatomy-and-physiology/> [Accessed Jan. 22, 2019].
- [3] P. Mayers and K. Grauer, "A Healthy 50-something with New Dyspnea on Exertion and an Interesting ECG," July 2017. [Online]. Available: <http://hqmeded-ecg.blogspot.com/2017/07/a-healthy-50-something-with-new-dyspnea.html> [Accessed Feb. 23, 2019].
- [4] R. Miramontes, R. Aquino, A. Flores, G. Rodríguez, R. Anguiano, A. Ríos, and A. Edwards, "Plaimos: A remote mobile healthcare platform to monitor cardiovascular and respiratory variables," vol.17, no.12, p.176, 2017.
- [5] R. Gandhi, "SVM model from scratch," June 2018. [Online]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> [Accessed Feb. 23, 2019].
- [6] H. Kandan, "Understanding the kernel trick," Aug. 2013. [Online]. Available: <https://towardsdatascience.com/understanding-the-kernel-trick-e0bc6112ef78> [Accessed Feb. 23, 2019].
- [7] "Confusion Matrix," 2016. [Online]. Available: https://www.scikit-yb.org/en/latest/api/classifier/confusion_matrix.html [Accessed Mar. 1, 2019].
- [8] "Cardiovascular diseases (CVDs)," May 2017. [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) [Accessed Feb. 14, 2019].

- [9] N. T. Srinivasan and R. J. Schilling, “Sudden cardiac death and arrhythmias,” vol.7, no.2, p.111, 2018.
- [10] “Sudden Cardiac Death (Sudden Cardiac Arrest),” [Online]. Available: <https://my.clevelandclinic.org/health/diseases/17522-sudden-cardiac-death-sudden-cardiac-arrest> [Accessed Jan. 21, 2019].
- [11] T. P. Exarchos, M. G. Tsipouras, C. P. Exarchos, C. Papaloukas, D. I. Fotiadis, and L. K. Michalis, “A methodology for the automated creation of fuzzy expert systems for ischaemic and arrhythmic beat classification based on a set of rules obtained by a decision tree,” vol.40, no.3, pp.187–200, 2007.
- [12] P. deChazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ecg morphology and heartbeat interval features,” vol.51, no.7, pp.1196–1206, 2004.
- [13] M. Llamedo and J. P. Martínez, “Heartbeat classification using feature selection driven by database generalization criteria,” vol.58, no.3, pp.616–625, 2011.
- [14] D. Zhang, chap. Wavelet Approach for ECG Baseline Wander Correction and Noise Reduction. IEEE, 2005.
- [15] Y. Bazi, N. Alajlan, H. AlHichri, and S. Malek, chap. Domain adaptation methods for ECG classification. IEEE, 2013.
- [16] E. R. Laskowski, “What’s a normal resting heart rate?,” Aug. 2018. [Online]. Available: <https://www.mayoclinic.org/healthy-lifestyle/fitness/expert-answers/heart-rate/faq-20057979> [Accessed Jan. 21, 2019].
- [17] “How the Heart Works,” [Online]. Available: <https://www.nhlbi.nih.gov/health-topics/how-heart-works> [Accessed Jan. 21, 2019].
- [18] “Overview of cardiac arrhythmias,” [Online]. Available: https://www.amboss.com/us/knowledge/Overview_of_cardiac_arrhythmias [Accessed Jan. 22, 2019].
- [19] R. N. Fogoros, “An Overview of Cardiac Arrhythmias,” Jan. 2019. [Online]. Available: <https://www.verywellhealth.com/overview-of-cardiac-arrhythmias-1746267> [Accessed Jan. 22, 2019].
- [20] E. J. d. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, “Ecg-based heart-beat classification for arrhythmia detection: A survey,” vol.127, pp.144–164, 2016.
- [21] “QRS Complex,” [Online]. Available: <https://www.healio.com/cardiology/learn-the-heart/ecg-review/ecg-interpretation-tutorial/qrs-complex> [Accessed Jan. 22, 2019].

- [22] “Kernel Functions-Introduction to SVM Kernel and Example,” Nov. 2018. [Online]. Available: <https://data-flair.training/blogs/svm-kernel-functions/> [Accessed Feb. 23, 2019].
- [23] A. KOWALCZYK, “Linear Kernel: Why is it recommended for text classification,” [Online]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> [Accessed Feb. 23, 2019].
- [24] J. Vert, K. Tsuda, and B. Schölkopf, *A Primer on Kernel Methods*, pp.35–70. Cambridge, MA, USA: MIT Press, 2004.
- [25] S. L. Joshi, R. A. Vatti, and R. V. Tornekar, chap. A Survey on ECG Signal Denoising Techniques. IEEE, 2013.
- [26] J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” vol.BME-32, no.3, pp.230–236, 1985.
- [27] T. Mar, S. Zaunseder, J. P. Martinez, M. Llamedo, and R. Poll, “Optimization of ecg classification by means of feature selection,” vol.58, no.8, pp.2168–2177, 2011.
- [28] Z. Zhang, J. Dong, X. Luo, K.-S. Choi, and X. Wu, “Heartbeat classification using disease-specific feature selection,” vol.46, pp.79–89, 2014.