



Machine Learning in Cardiac Health Monitoring and Decision Support

Shrouq Hijazi and Alex Page, University of Rochester

Burak Kantarci, University of Ottawa

Tolga Soyata, SUNY Albany

Portable medical devices generate volumes of data that could be useful in identifying health risks. The proposed method filters patients' electrocardiograms (ECGs) and applies machine-learning classifiers to identify cardiac health risks and estimate severity. The authors present the results of applying their method in a case study.

As personalized medicine becomes increasingly more sophisticated and affordable, portable medical devices have become ubiquitous and monitoring applications have begun to blend a range of functions. AliveCor, for example, offers an inexpensive smartphone electrocardiogram (ECG) attachment that can sample an individual's ECG, calculate real-time statistics, and share the recording with a physician.

In aggregate, portable medical devices generate data at a much higher rate than conventional systems, which can overwhelm medical personnel who must review reports for many patients. However, the data also presents scientists and engineers with the opportunity to create health-monitoring and decision-support systems that enhance and personalize healthcare. For example,

decision-support systems based on machine learning (ML) can ease the review burden by filtering noise, errors, and irrelevant information so that the data reviewed contains only relevant clinical markers. ML algorithms learn patterns within the data, which serve as the basis for predictions about patient health. A machine can look through millions of reports and medical records to identify previously unknown drug interactions.¹ Such algorithms can significantly improve diagnostic accuracy, healthcare quality, and patients' quality of life.

ML algorithms have many potential applications in smart health. To explore one of these, we developed a method that filters data from long-term ECG recordings of patients with Long QT Syndrome (LQTS), uses ML to identify circadian patterns that signal risk of symptoms such as cardiac arrhythmia, and estimates the

severity of that risk. LQTS is a disorder that primarily affects ion channels in heart muscle cells, allowing abnormal electrical activity to occur that can lead to sudden and dangerous arrhythmias. We tested our method using four classification methods against a database of 434 24-hour ECG recordings. We also explored how our method might scale with the volume of medical data. Scalability is rapidly becoming critical in medical studies. Analyzing massive amounts of data not only promotes a deeper understanding of the mechanisms that cause diseases but also makes personalized treatment possible. Such analyses can lead to breakthroughs in relating genes to diseases as well as providing the basis for treatment oriented to a particular patient's lifestyle and genetic makeup.

AN ML-BASED SYSTEM

We envision incorporating our method in a remote health-monitoring system that can provide feedback and decision support to a clinician. The system would use devices that acquire data through the Internet of Things and are connected to a cloud-based decision-support system.² The technological components of such a system are within reach, and advanced devices for acquiring medical data are becoming commercially available.³ Sophisticated and powerful ML algorithms are already well understood and accessible.⁴

However, the human brain has unmatched reasoning abilities, so the physician is still the most important part of any medical decision-support system. Thus, the goal of our envisioned system is to provide physicians or other clinicians with concise, relevant information that can increase their diagnostic efficiency and accuracy.

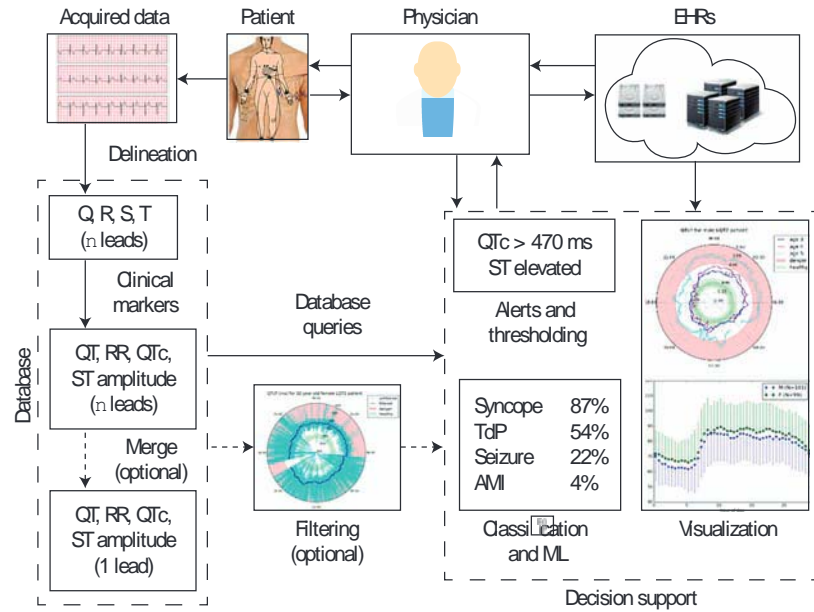


FIGURE 1. Conceptual workflow of a remote health-monitoring and decision-support system. Data from a patient at a remote location is acquired, preprocessed, and used to provide decision support in several forms. Visualization provides simple summaries to the physician or other clinician without any recommendations. Alerts are triggered by more urgent events, such as the violation of an established threshold. Classification of the patient's condition is based on the results of machine learning (ML), which involves comparing the patient to existing electronic health records (EHRs). AMI: acute myocardial infarction; Q, R, S, T: waves that indicate cardiac electrical state (on an electrocardiogram [ECG]); QT, RR, QTc, and ST: intervals in the cardiac electrical cycle, also measured on an ECG; TdP: torsades de pointes, a cardiac arrhythmia.

Workflow

Figure 1 is a conceptual diagram of the workflow for a healthcare system that stores patient data electronically. After preprocessing and filtering patient data, the system stores it as an electronic health record (EHR). Each EHR gradually enriches the database, which will improve the accuracy of future ML results. A large database with many patients' records might not be as useful as a database with fewer patients but more information on each patient.

When many patients' records are aggregated and analyzed, steps must be taken to protect the individuals' privacy. Most protected health information (PHI), such as names and birthdays, can be removed from records in compliance with the Health Insurance Portability and Accountability Act (HIPAA) without detriment to the data mining process.⁵ However, in some

cases, it would be desirable to obtain more detailed information about certain patients from their physicians—an impossibility because it would violate HIPAA. Consequently, regulation, not just technology, can limit the acquisition of needed data.

Even after removing identifying information, what remains could be combined to statistically reveal a patient's identity. On one hand, PHI information such as age, gender, race, and genetic disorders, is critical to developing an effective decision-support system. On the other, including too much information on the wrong computer system risks violating HIPAA. Applications can also create privacy violations because some require explicitly protected information such as a patient's voice print⁶ or city of residence. Researchers must keep these restrictions in mind during all stages of a study.

Decision support

An effective ML-based healthcare system capitalizes on the computer's vast computational capability and the physician's reasoning ability. Both machine and physician are looking for patterns, but the physician cannot analyze every heartbeat of every patient or be familiar with every disease's nuances. The machine can do all these tasks and then present its conclusions to the physician for confirmation.

Support types. As Figure 1 shows (see box at lower right), we envision three types of decision support: visualization, alerts, and classification.

Visualization puts long-term monitoring data in a concise and intuitive format,⁷ which could significantly reduce the physician's data burden and enable timely and accurate decision making.

Alerts are alarms that activate when a value crosses an established threshold. The value can be simple to check, or the result of a more advanced algorithm. The threshold could be a clinical standard—for example, 480 ms for QTc, which is a measure of the ventricular depolarization and repolarization duration—or it could be tailored to a patient. For example, the physician might want to be notified only if a particular patient's QTc exceeds 600 ms.

Classification is the process of predicting the group a patient belongs in, such as people with a specific genotype or people at risk for certain cardiac events. Prediction of short-term outcomes is a primary goal. For example, the machine might predict that a patient is at high risk for a myocardial infarction in the next 12 hours.

The outputs from these support systems, such as plots and recommendations, would be attached to the

typical ECG report that a cardiologist reviews. When real-time monitoring reveals an urgent issue, an alert would immediately be sent to both the patient and physician through SMS, pager, or an application.

Evolving symbiosis. The physician is still at the head of this process—ordering tests, analyzing records, adjusting prescriptions, and so on. The machine's visualizations and recommendations are simply additional decision-making tools. Over time, the database will expand, and the machine's classifications will be more accurate. But improvements will be symbiotic: as the machine's accuracy grows, the physician will develop an intuition for how and when the machine makes those accurate classifications and recognize when it might be fallible. For example, a patient might have an abnormal T-wave morphology that the algorithms did not process correctly, or a patient's heart rate had not reached the point at which problems would be identifiable. The physician can recognize the machine's limitations, and might opt for additional methods to measure risk such as prescribing a drug or exercise challenge or conducting a manual ECG analysis.

CASE STUDY PARAMETERS

In a case study to evaluate our method, we exploited ML's pattern-recognition abilities to classify risk in LQTS patients. The QT interval, which is typically used to measure the duration of ventricular repolarization (a clinical marker of the heart's electrical activity), can be abnormally long in some people who are taking certain medications or have certain genetic disorders.⁸ A prolonged QT interval can trigger arrhythmias such as torsades de

pointes (TdP), which are likely to cause serious symptoms such as seizures, fainting, or sudden death. It is therefore critical to monitor the QT interval in patients prone to this disorder using long-term ECG recordings. Data recordings of ambulatory patients over several hours or days are called Holter recordings or simply Holters.

Our study focused on congenital LQTS rather than the drug-induced form. In the database used for the study, we knew which recordings were from patients with symptoms, but we did not know if the symptoms came before or after the ECG recording. Consequently, we had no way to use ML to predict when symptoms would occur or to detect symptoms in real time. Instead, we attempted to identify when a recording came from a patient whom we knew had symptoms in the past or would have them in the future. In other words, we attempted to identify the patient's risk—an important concern for physicians, who must often prescribe medications and implantable devices on the basis of the perceived risk of symptoms. (Symptoms in this context are events, such as syncope, that are triggered by prolonged QT.)

Identifying high-risk patients who need extra prescriptions or monitoring or low-risk patients who would not benefit from those burdens would be highly valuable to both the physician and patient. Additionally, despite the limitations of this particular database, this study laid the groundwork for a future study at a time when a dataset with clinical outcomes becomes available.

Data preprocessing

Figure 2 illustrates the steps that transform raw ECG data into clinically useful measurements. Raw data

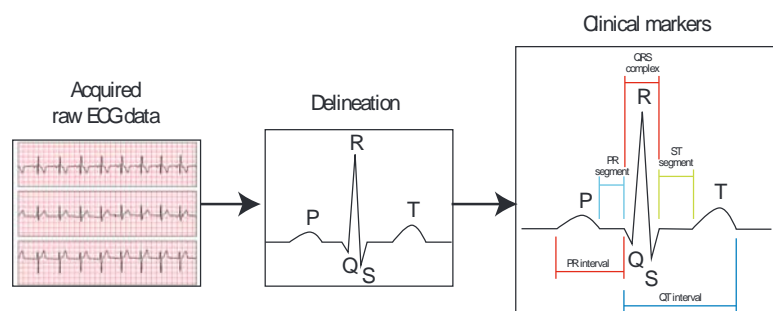


FIGURE 2. Detailed preprocessing steps using the workflow from Figure 1. The raw ECG signal contains too much data to feed into most ML algorithms, and the data has massive redundancy across leads (sensor locations) and heartbeats, as well as noise and errors. Preprocessing filters the raw ECG waveforms to extract only relevant clinical markers, such as the durations of the QRS complex and P and T waves.

contains massive redundancies as well as ectopic heartbeats and obvious noise and errors, which will not be useful in later processing. Preprocessing extracts only clinically relevant markers from the data, which reduces the data fed into the ML algorithm by multiple orders of magnitude while drastically improving classification accuracy and execution time. The delineated heartbeat (Clinical markers box) highlights several important measurements of the cardiac electrical cycle. Atrial and ventricular depolarization and repolarization are represented on the ECG as a series of waves: the P wave followed by the QRS complex and the T wave. The P wave occurs during atrial depolarization—when the atria contract. The QRS complex indicates ventricular depolarization—when the ventricles contract; at its end is the J point. The T wave occurs during ventricular repolarization—when the ventricles relax. Durations, amplitudes, and shapes at various parts of the ECG can be used to diagnose myriad illnesses.

Algorithm training

To identify patients at risk for LQTS symptoms, we trained ML algorithms using input variables extracted from raw ECG data. As Figure 2 shows, useful intervals and amplitudes are available after only two preprocessing steps: delineation and the computation of clinical markers.

We first annotated important markers in the ECG recording (such as the P, Q, R, S, and T peaks, onsets, and offsets) using delineation software to identify these points.^{9,10}

The relevance of each clinical marker depends on the disease being studied. For example, STe, the elevation (voltage) during the ST segment,

is of interest in heart-attack cases, and the shape of the P wave is of interest to diagnosing atrial enlargement. Many diseases affect the QT interval and its subintervals (QRS, the J point to T peak, and T peak to T end).

In our study, the QT and RR intervals were the most relevant markers. The QT interval alone is not enough information, because it will naturally lengthen and shorten in all individuals as their heart rate decreases or increases. The RR interval—the duration of a complete cardiac cycle—provides enough information to correct the QT for heart rate. We calculated the corrected QT (QTc) using the Fridericia equation:¹¹

$$QTc = \frac{QT}{\sqrt[3]{\frac{RR}{s}}}$$

Reducing dimensionality

Preprocessing substantially reduces the data to be reviewed. The raw ECG data—sampled at 200 Hz, 16 bits per sample, on 3 leads, and over 24 hours—needed 100 Mbytes of storage. If only QTc interval values are required to detect LQTS symptoms, the storage requirement lowers to 1 Mbyte. Storage capacity alone is not a sufficient reason to have preprocessing; additional storage space is relatively inexpensive. Rather, preprocessing is necessary when using ML algorithms because the data reduction translates directly to a dimensionality

reduction and faster processing in subsequent steps.

Reduced dimensionality is important because the “curse of dimensionality” remains a difficult problem in categorizing big data. That is, large volumes of data have a daunting number of features, with each feature having myriad possible values. There are so many dimensions to work with that it is too easy to separate the training data into the correct groups. The learned model becomes specific to the training set rather than generalizing to other data—a problem known as overfitting. Thus, an enormous amount of training data is required to ensure that there are several samples with each combination of values. With a fixed number of training samples, ML’s predictive power can decrease as dimensionality increases.

Method selection

Among the ML methods for classification, supervised learning and clustering are the most popular. In our study, we focused on supervised learning. The alternative to supervised learning is clustering, also known as unsupervised learning, which generally tries to group data points into clusters according to their proximity to one another. A new data point can then be classified on the basis of the cluster into which it best fits.

Artificial neural networks, inspired by the neuron web in the human brain, can be used for both supervised and

TABLE 1. Pros and cons of four supervised learning classifiers.

Classifier	Advantages	Disadvantages
k-nearest neighbors	Simple to implement Easy to understand and interpret	Sensitive to noisy data and anomalies Computationally expensive for large datasets
Support vector machine	Flexible with nonlinear data Scales up with large sets of data Relatively resistant to the “curse of dimensionality”	Difficult to interpret feature importance Yields possibly unreliable confidence estimates
Random forest	Alleviates overfitting problem Easy to extract feature importance Resilient to missing data Scales to large datasets	Increases bias relative to single decision tree Different results possible in retraining on same data
AdaBoost	Automatically reduces dimensionality Relatively fast	Sensitive to noisy data and anomalies

unsupervised learning. Deep networks use many layers of artificial neurons to form input data abstractions, which lead to the formation of a final classification layer. In each layer, weights are applied to features of the previous layer to optimize performance.

The supervised learning methods that best fit our study were support vector machine (SVM), decision tree, and nearest neighbors. These algorithms classify previously unseen data points based on some function of the data points they have already seen. In an attempt to improve accuracy, some ML algorithms factor in the results of several classifiers in order to make their decision. Random forest and AdaBoost are among the most popular methods to use this ensemble learning technique.

Table 1 lists the pros and cons of the four methods we considered in our evaluation: k-nearest neighbors, SVM, random forest, and AdaBoost. The table is useful in identifying the best classifier for a given problem and set of computational constraints.

k-nearest neighbors. The k-nearest neighbors algorithm finds the shortest distance between a new testing point and adjacent training points. It then classifies the testing point as the most common class among its k-nearest neighbors.

Support vector machine. The SVM method uses a training points subset

to create hyperplanes that divide the data into classes, which it keeps as far apart as possible (thereby maximizing the distance between the hyperplane and different class samples). SVMs rely on a linear or nonlinear feature combination, depending on the kernel declared in the algorithm. We tested a linear kernel as well as a radial basis function (RBF) kernel.

Random forest. The random forest algorithm is an ensemble learning method that averages the results of several decision trees to classify its samples. As the name implies, each decision tree is trained on a random training data subset, perhaps using random features as well.

AdaBoost. AdaBoost, short for adaptive boosting, aggregates the results from many weak classifiers by iteratively retraining them to focus on fixing mistakes from the previous round. It then averages the results. In our experiments, AdaBoost always used decision trees as the weak classifier.

RUNNING THE CLASSIFIERS

We accessed a database containing 24-hour ECG recordings of 480 LQTS patients, including demographic information such as gender, age, and specific LQTS genotype.¹² We restricted our study to 434 recordings of patients with the most common

LQTS genotypes (LQT1 and LQT2) and the most complete demographic information (such as age and gender). The subjects’ average age was 25 ± 18 years (newborns to senior citizens); 55 percent of the subjects were female, and 67 percent had the LQT1 mutation. Our goal was to determine which of the patients would show symptoms of LQTS, such as seizures or syncope. That is, we were trying to identify ECG patterns that could reveal which genotype-positive patients will also be phenotype-positive. Given some measurements from an ECG, a classifier should simply tell us “symptoms expected” or “no symptoms expected,” perhaps with a confidence value.

Algorithm implementation

We implemented all the classification algorithms—k-nearest neighbors, linear SVM and RBF SVM, random forest, and AdaBoost—using scikit-learn, an open source Python library built on SciPy and NumPy.⁴ To assess a classifier’s accuracy, we set aside 30 percent of the samples for testing, and trained only on the remaining 70 percent. Because some algorithms include inherent randomness in their operation and because the division between training and testing data is also random, we repeated the cycle of selecting training data, training, and testing 50 times for each classifier.

The average result from these trials—the Monte Carlo cross-validation

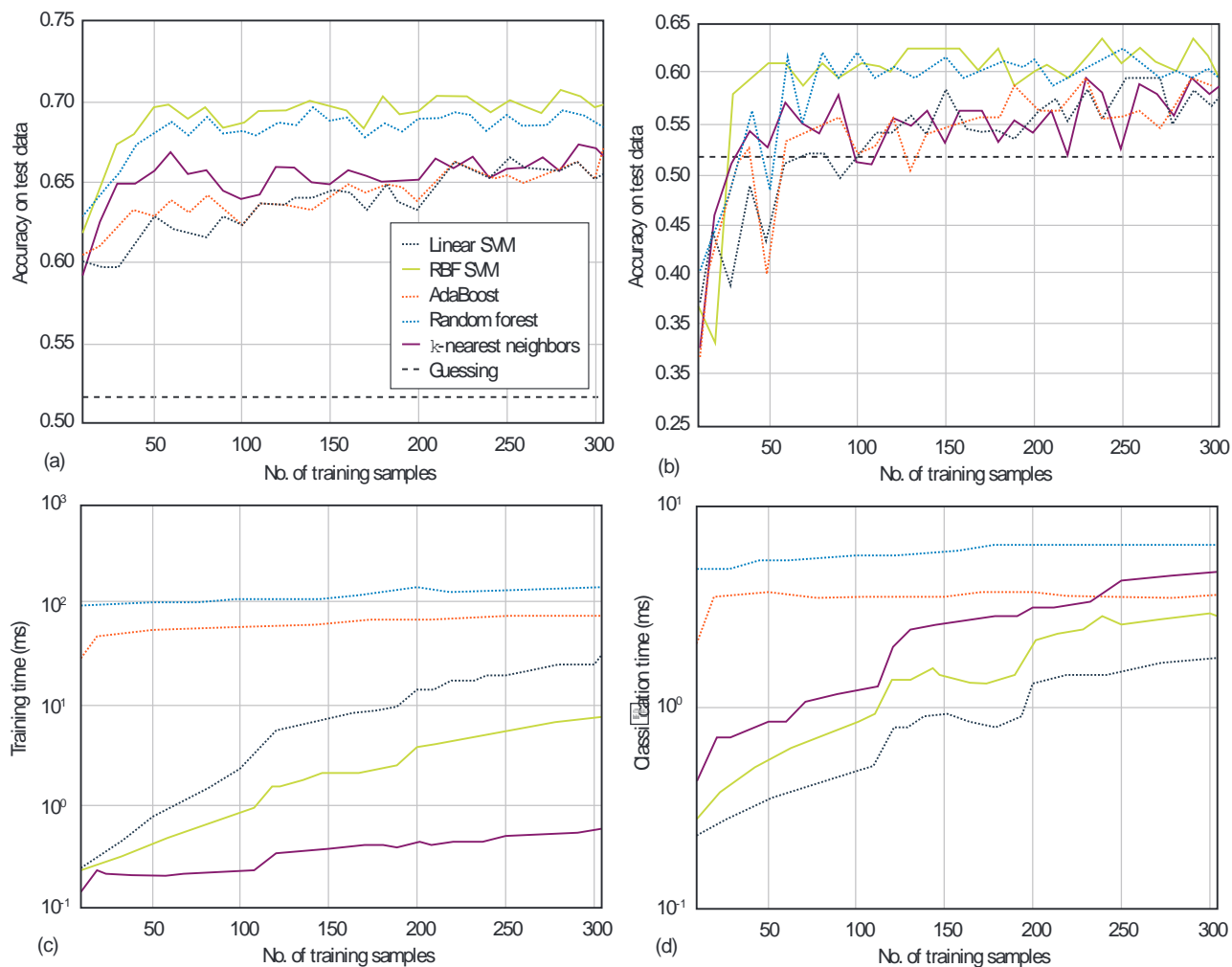


FIGURE 3. Classifier performance in the study. The input features were hourly QT and RR measurements for one day. Each data point is an average of 50 trials for which we randomly selected different training and testing data. Approximately 52 percent of the patients did not have symptoms, so the thin dashed line in (a) average accuracy and (b) minimum accuracy represents the performance achievable by simply guessing “no symptoms” every time. (c) Training time represents training conducted for increasingly larger subsets of the full training dataset of 304 samples, and (d) classification time represents how long validation took for the full test set of 130 samples. AdaBoost always used decision trees as the weak classifier.

score—told us how well a classifier would likely perform. The worst result showed how low a classifier’s accuracy could be when the data was not evenly distributed across the random training/testing split (based on its underlying properties). For example, if most of the outliers or noisy recordings end up in the test set, or almost all of the asymptomatic patients end up in the training set, our trained model will not match up well with the data it’s tested against. For classifiers that require input data to be normalized, we used scikit-learn’s `StandardScaler()` function.

Feature selection

The four ML methods we used have inherent strengths and weaknesses, but their performance was also constrained by the data we provided them. We knew that QTc, and therefore QT and RR, are the measurements that cardiologists use most often to determine whether a LQTS patient is in danger. We also knew that people with different LQT genotypes tend to show more QTc prolongation at different times of the day.¹³ We therefore decided to provide hourly QT and RR measurements as input to the ML

classifiers. Each of our samples for training or classification consisted of 48 values (24 for QT and 24 for RR); increasing that number risked in fitting the curse of dimensionality.

To reduce dimensionality even more, we used chi-square (χ^2) tests to automatically select features that were likely to be the most useful. In general, the fewer the dimensions in the input, the fewer training samples are needed to achieve good performance, and the faster classifiers will run. Feature selection methods are also useful in the discovery of previously unknown

patterns and correlations between variables. For example, the machine might find that patients with a very specific genetic mutation are more at risk than others with seemingly similar mutations or that a measurement that is typically not used in the clinic actually carries significant information. Even if no new relationships are discovered, feature selection is useful to confirm the chosen model—to see if the machine picks the features expected.

RESULTS

We used the four classifiers to determine if the genotype-positive LQTS patients in our database had suffered or would suffer from any symptoms. Figure 3 illustrates how the performance of each classifier changes as we provided more training samples. We configured the k-nearest neigh-

testing. However, we conducted training over increasingly larger subsets of the remaining 70 percent to determine how many samples would produce optimal results.

Average and minimum accuracy

Figure 3a shows average accuracy—the accuracies that we could expect from each classifier on the basis of 50 random training-data selections. Figure 3b shows minimum accuracy over the 50 trials—assuming we chose training data poorly, how well could each algorithm do? We found that in this worst-case scenario, more than 100 training samples could be required simply to break even—to exceed 52 percent (the guessing line). The highest scores, both minimum and average, came from random forest and the RBF SVM, which achieved 60 to 65 percent accuracy even with a poor selec-

tion of data and fewer than 100 training samples. The best classifier in our tests, the RBF SVM, averaged about 70 percent accuracy.

Scalability

Figures 3c and 3d show the results of measuring the runtime of the training and classification stages, which we used to estimate each classifier’s scalability. Runtime was not a problem with this particular dataset, but it could be a limitation in other studies. Although the ensemble classifiers (AdaBoost and random forest) took longer than the others in both stages, adding training samples barely affected their runtimes. As we expected, k-nearest neighbors had essentially zero training time, but classification time increased with the number of samples because the algorithm had to compute distances to every point in the training set. In fact, at around 240 training samples, classification actually became slower than with AdaBoost. Because the two ensemble methods have very fast runtimes, SVM will also become slower than even random forest, given enough training samples.

All the classifiers except k-nearest neighbors (which does not really have a training stage) could not incorporate new data after training was complete. When a database grows, classifiers must be entirely retrained or the process of adding samples must use nontrivial techniques. The ability to add one or more training examples to a model without complete retraining, referred to as online ML, is an

IN FEATURE SELECTION, OUR AIM WAS TO REDUCE INPUT DIMENSIONALITY WHILE MINIMIZING RELEVANT INFORMATION LOSS.

bors classifier to weight samples by distance rather than uniformly and composed the random forest with 100 trees rather than the default of 10. We set both random forest and the two SVMs to use balanced class weights. All other parameters were the scikit-learn defaults. Each of the classifiers’ input samples contained 48 values.

We computed runtime results on an Intel i7-5930K and always used 30 percent of the 434 samples in the full dataset (training plus testing) for

tion of data and fewer than 100 training samples. The best classifier in our tests, the RBF SVM, averaged about 70 percent accuracy.

Obviously, it is not desirable for the machine to make bad classifications. However, relatively low accuracy is manageable if we know when the computer was unsure of a result. We therefore tested the machine’s average confidence in its responses. When the computer was incorrect, its average confidence was around 64 to 69

important scalability feature. Consequently, we tested the perceptron online algorithm¹⁴ and found that it achieved 68 percent accuracy, a level comparable to that of random forest.

Effects of feature reduction

We next investigated the impact of providing classifiers with QTc alone instead of QT and RR separately, which reduced the number of features from 48 to 24. Because QTc is designed to contain the LQTS-relevant information from QT and RR, we expected to see improved runtimes without lower accuracy. However, we found that accuracy decreased for random forest and AdaBoost (both of which are based on decision trees) as did the learning rate (more samples were required to reach peak performance). The feature reduction did not hurt RBF SVM's performance, which consistently yielded 70 percent accuracy when provided with enough training samples.

AdaBoost and random forest saw no improvement in runtime with fewer features. Predictably, k-nearest neighbors' classifications were faster, and all SVM runtimes improved. Because of these results, we revisited our feature choice, attempting to narrow the list as much as possible.

Searching for better features

Our use of QT, RR, and QTc was based on knowing what physicians measure in practice. However, we wanted to be sure that we did not overlook any other useful cardiac features, so we decided to evaluate 23 features at every hour of the day—a total of 552 measurements that included QT, RR, QRS, ST segment duration and elevation, QTp, JT, JTp, TpTe, and T-wave duration and amplitude. Additionally, we used several features from

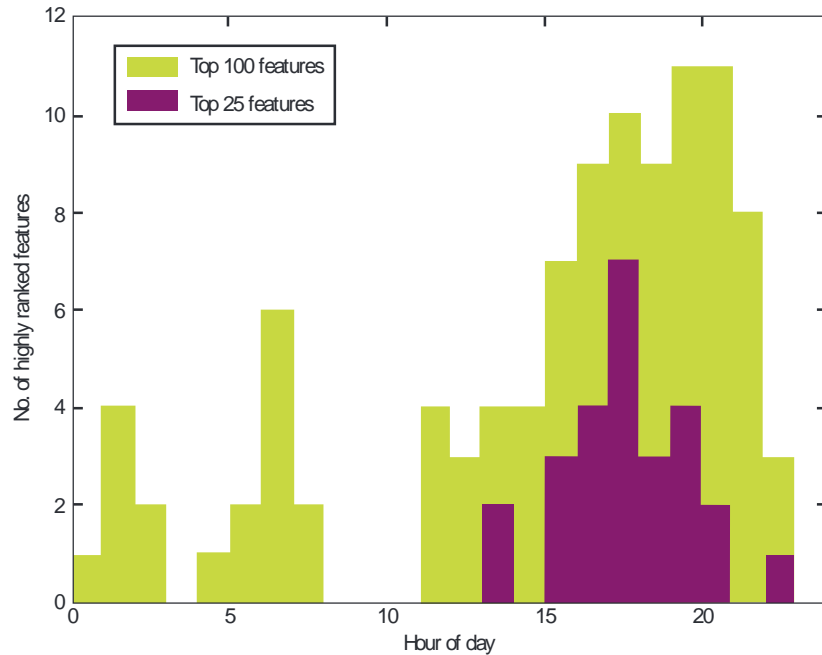


FIGURE 4. Feature importance versus time of day. Starting at 0, which represents midnight, the peaks indicate that late night, awakening, and the end of the workday are the best times to detect cardiac issues for this patient group.

each patient's EHR: gender, age, LQT type (1 or 2), mutation type, and mutation location.

Because we had only 434 training samples, we expected to have to reduce the new feature set's dimensionality to mitigate overfitting. Our aim was to reduce input dimensionality while minimizing information loss, which both the principal component analysis (PCA) and the χ^2 method satisfy. PCA projects the data to a lower dimensional space, while χ^2 selects the statistically best features. We measured classifier accuracy varying the number of preserved features from 1 to 512. Surprisingly, both feature selection methods allowed the classifiers to achieve 70 percent accuracy with only one feature or attribute. The most important features seemed to be QT-like measurements taken in the evening, where "QT-like" means QT, JT, QTp, JTp, or versions of these corrected for heart rate. Using the top 20 features with the random forest classifier yielded 72 percent accuracy (69 percent sensitivity and 75 percent specificity).

Figure 4 is a histogram of the times of day in which the top 25 and top 100 features appeared. As the figure shows, all the top 25 features are evident around 5 pm to 6 pm, which implies that perhaps fatigue at the end of the workday is unmasking cardiac issues. Expanding the search to the top 100 features begins to reveal other important times. One is first thing in the morning (6 am to 7 am), which is another stressful time of day.¹⁵ Another is late night (1 am to 2 am), which makes sense for the LQT2 subset of patients who tend to show more QTc prolongation during sleep.¹³ Also important is the lack of highly ranked features around 8 am to 11 am, which implies that a clinical checkup in the morning might not be sufficient for the physician to accurately assess a patient's risk.

STUDY IMPLICATIONS

Our study had several implications for future analyses, including the effects of beta blockers, gender influence on classification, and measurement type.

Beta blocker effects

In one of our experiments, we found that separating QTc into QT and RR improved accuracy. However, many high-risk LQTS patients are on beta blockers—drugs prescribed to slow the heart rate—so it was possible that classifiers in that experiment were actually reacting to the increased RR

patients had symptoms. When combined, the line was 52 percent.

Our second experiment was to train the classifier on both groups as before, but to test the accuracy against each group separately. Results showed little difference: classification of only BB or only non-BB patients remained at 66 to 68 percent accuracy.

virtually no weight placed on that feature. It will be interesting to see under what conditions gender or the other static inputs become significant.

Measurement type

For our study, the feature of interest (QTc) has characteristics that fit well with hourly average measurements because QTc changes slowly and is corrected for heart rate. However, for other applications, even the extensive feature set we tested might not be enough. Heart rate, for example, can vary greatly during one hour. Perhaps a different measure such as heart rate variability would be more suitable. Future analyses might even include more exotic measurements, such as “ST elevation at 60 ms after J point during high heart rate” or “percentage of beats that T wave is inverted.”

THE ML SYSTEM WE ENVISION WILL PROVIDE A PERSONALIZED ASSESSMENT OF EACH PATIENT BASED ON A COMBINATION OF CRITICAL MARKERS.

that is characteristic of beta blockers, not to any novel pattern we had hoped to find. In particular, beta blockers might explain why the classifiers based on decision trees (AdaBoost and random forest) had higher accuracy when heart rate was available. To determine if this was the case, we conducted three evaluations.

The first was to train the classifiers on only beta blockers (BBs) or non-BB patients and then check accuracy. When the classifiers were trained on only BB patients, overall accuracy remained at approximately 70 percent. Accuracy within the BB group increased to approximately 90 percent, but accuracy in the non-BB group fell to approximately 60 percent. When we trained the classifier on only the non-BB group, we observed the opposite results. Although 90 percent accuracy is a marked improvement, the “guessing” line is higher in these subgroups: 67 percent of non-BB patients had no symptoms, and 62 percent of BB

Finally, in our experiment that used only QTc as input, we were not (ideally) providing any heart-rate information to the classifier. The classifier needed more training samples to reach peak accuracy, but that peak was still around 70 percent.

We concluded that the presence of BBs does not affect overall accuracy, but classifiers are more accurate when the groups are separated.

Gender

We expected gender to be a feature of significant importance in ML classification, as gender differences are known to influence many coronary heart diseases,¹⁶ and males and females have distinct average QTc values and clinical prolongation thresholds. However, feature selection eliminated gender (along with age and mutation information) as very insignificant relative to most ECG measurements. We confirmed this by running RBF SVM and random forest with gender as an input, and found no difference in results and

We have presented a workflow and a conceptual ML-based system for health monitoring that aims to analyze the ECGs of patients with an LQTS genetic disorder and to identify patients with increased risk of adverse cardiac events. The envisioned system will provide a personalized assessment of each patient by considering a combination of critical markers.

The results in Figure 3 provide insights into which classifier works best under constraints such as available training data or computational power. RBF SVM or random forest will yield the highest accuracy. RBF SVM is probably the better choice for experiments with relatively few training samples, but random forest’s faster runtime will be preferable when training data grows to thousands of samples. As runtime

becomes more of a concern, feature selection gets more important.

In Figure 3a, all classifiers seem to be close to reaching a horizontal asymptote, meaning that their performance will not improve simply by adding more training samples. Instead, their inputs and parameters will need to be optimized. In previous work, we found that time of day was important in classifying patients as having the LQT1 or LQT2 genotype.^{13,17} Both types show QT prolongation, but during different activities and different times of day. This finding is in large part why we structured our inputs as hourly data points in the study described. The dimensions also reduce well in this structure, as usually only a few hours during sleep are enough to differentiate LQTS types. However, other input structures and measurements should be investigated. In the selection of appropriate input features, the physician's knowledge and intuition remain critical.

The annotation algorithm we used had some trouble with noisy or abnormal ECGs; improved accuracy might require cleaner inputs and more accurate annotations. Additionally, we could construct more complex features such as T-wave symmetry measurements. Another possible approach is the use of a voting classifier, which attempts to aggregate the predictions of several other classifiers to reach a better result. However, our experience suggests that a voting classifier will be only slightly more accurate than the best individual classifier. Finally, a complete set of experiments will require trying other classification methods such as clustering and artificial neural networks.

The steps we used can be generalized to other types of medical data and

illnesses. We expect that the refinement of our method and the growth of EHR databases will greatly improve the quality of care for patients with a variety of disorders. ■

ACKNOWLEDGMENTS

We thank Mehmet Aktas and Jean-Philippe Couderc from the University of Rochester's Department of Medicine for motivating this study and providing guidance about its clinical applications. This work was supported in part by National Science Foundation grant CNS-1239423.

ABOUT THE AUTHORS

SHUROUQ HIJAZI is a technology consultant at Ernst & Young. While conducting the research reported in this article, she was a research assistant in the machine-learning (ML) laboratory at the University of Rochester. Her research interests include ML techniques, cybersecurity, computer networks, the Internet of Things (IoT), and virtualization. Hijazi received a BS in electrical and computer engineering from the University of Rochester. She is a student member of IEEE. Contact her at shijazi@u.rochester.edu.

ALEX PAGE is a postdoctoral associate in the Heart Research Follow-up Program at the University of Rochester Medical Center. While conducting the research reported in this article, he was a PhD student in electrical engineering at the University of Rochester. His research interests include computer systems for analyzing medical data, such as databases, GPU acceleration, and ML techniques. Page received a PhD in electrical engineering from the University of Rochester. He is a student member of IEEE. Contact him at alex.page@rochester.edu.

BURAK KANTARCI is an assistant professor in the School of Electrical Engineering and Computer Science at the University of Ottawa and a courtesy assistant professor in the Electrical and Computer Engineering Department at Clarkson University. His research interests include the IoT, big data in the network, crowdsensing and social networks, cloud networking, and digital health. Kantarci received a PhD in computer engineering from Istanbul Technical University. He is an editor of IEEE Communications Surveys and Tutorials, a Senior Member of IEEE, and a member of ACM. Contact him at burak.kantarci@uottawa.ca.

TOLGA SOYATA is an associate professor in the Department of Electrical and Computer Engineering at SUNY Albany. His research interests include cyber-physical systems, digital health, and GPU-based high-performance computing. Soyata received a PhD in electrical and computer engineering from the University of Rochester. He is a Senior Member of IEEE and ACM. Contact him at tsoyata@albany.edu.

REFERENCES

1. T. Lorberbaum et al., "An Integrative Data Science Pipeline to Identify Novel Drug Interactions That Prolong the QT interval," *Drug Safety*, vol. 39, no. 5, 2016, pp. 433–441.
2. M. Hassanaliheragh et al., "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges," *Proc. IEEE Int'l Conf. Services Computing (SCC 15)*, 2015, pp. 285–292.
3. D. Son et al., "Multifunctional Wearable Devices for Diagnosis and

Therapy of Movement Disorders," Nature Nanotechnology, vol. [1], pp. [1].

[1] F. Pedregosa et al., "Scikit-Learn: Machine Learning in Python," J. Machine Learning Research, vol. [1], pp. [1].

[1] "Summary of HIPAA Privacy Rule," US Department of Health and Human Services, [1], www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations.

[1] E.C. Larson et al., "Spirosmart: Using a Microphone to Measure Lung Function on a Mobile Phone," Proc. ACM Conf. Ubiquitous Computing (UbiComp [1]), pp. [1].

[1] A. Page et al., "An Open Source ECG Clock Generator for Visualization of Long-Term Cardiac Monitoring Data," IEEE Access, vol. [1], pp. [1].

[1] C.E. Chiang, "Congenital and Acquired Long QT Syndrome: Current Concepts and Management," Cardiology Rev., vol. [1], no. [1], pp. [1].

[1] Y. Chesnokov, D. Nerukh, and R. Glen, "Individually Adaptable Automatic QT Detector," Proc. IEEE Computers in Cardiology (CinC [1]), pp. [1].

[1] A. Demski and M.L. Soria, "Ecg-Kit: A Matlab Toolbox for Cardiovascular Signal Processing," J. Open Research Software, vol. [1], no. [1], openresearchsoftware.metajnl.com/articles/[1].ors.[1].

[1] L.S. Fridericia, "The Duration of Systole in an Electrocardiogram in Normal Humans and in Patients with Heart Disease," vol. [1], pp. [1] (in German).

[1] J. Couderc, "The Telemetric and Holter ECG Warehouse Initiative (THEW): A Data Repository for the Design, Implementation and Validation of ECG-Related Technologies," Proc. IEEE Int'l Conf. Eng. Medicine and Biology Soc. (EMBC [1]), pp. [1].

[1] A. Page et al., "QT Clock to Improve Detection of QT Prolongation in Long QT Syndrome Patients," Heart Rhythm, vol. [1], no. [1], pp. [1].

[1] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Rev., vol. [1], no. [1], pp. [1].

[1] W.B. White, "Cardiovascular Risk and Therapeutic Intervention for the Early Morning Surge in Blood Pressure and Heart Rate," Blood Pressure Monitoring, vol. [1], no. [1], pp. [1].

[1] A. Maas and Y. Appelman, "Gender Differences in Coronary Heart Disease," Netherlands Heart J., vol. [1], no. [1], pp. [1].

[1] A. Page et al., "Research Directions in Cloud-Based Decision Support Systems for Health Monitoring Using Internet-of-Things Driven Data Acquisition," Int'l J. Services Computing, vol. [1], no. [1], pp. [1].




2017 B. Ramakrishna Rau Award Call for Nominations

Honoring contributions to the computer microarchitecture field

New Deadline: 1 May 2017



Established in memory of Dr. B. (Bob) Ramakrishna Rau, the award recognizes his distinguished career in promoting and expanding the use of innovative computer microarchitecture techniques, including his innovation in compiler technology, his leadership in academic and industrial computer architecture, and his extremely high personal and ethical standards.

WHO IS ELIGIBLE? The candidate will have made an outstanding innovative contribution or contributions to microarchitecture, use of novel microarchitectural techniques or compiler/architecture interfacing. It is hoped, but not required, that the winner will have also contributed to the computer microarchitecture community through teaching, mentoring, or community service.

AWARD: Certificate and a \$2,000 honorarium.

PRESENTATION: Annually presented at the ACM/IEEE International Symposium on Microarchitecture

NOMINATION SUBMISSION: This award requires 3 endorsements. Nominations are being accepted electronically: www.computer.org/web/awards/rau

CONTACT US: Send any award-related questions to awards@computer.org

www.computer.org/awards

myCS Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>.