J Healthc Inform Res (2018) 2:179–203
https://doi.org/10.1007/s41666-018-0021-1

RESEARCH ARTICLE

Machine Learning and Mobile Health Monitoring Platforms: A Case Study on Research and Implementation Challenges

Omar Boursalie[1] · **Reza Samavi**[2,3] · **Thomas E. Doyle**[1,3,4]

Received: 15 June 2017 / Revised: 12 April 2018 / Accepted: 17 April 2018 /
Published online: 22 May 2018
© Springer International Publishing AG, part of Springer Nature 2018

Abstract Machine learning-based patient monitoring systems are generally deployed on remote servers for analyzing heterogeneous data. While recent advances in mobile technology provide new opportunities to deploy such systems directly on mobile devices, the development and deployment challenges are not being extensively studied by the research community. In this paper, we systematically investigate challenges associated with each stage of the development and deployment of a machine learning-based patient monitoring system on a mobile device. For each class of challenges, we provide a number of recommendations that can be used by the researchers, system designers, and developers working on mobile-based predictive and monitoring systems. The results of our investigation show that when developers are dealing with mobile platforms, they must evaluate the predictive systems based on its classification and computational performance. Accordingly, we propose a new machine learning training and deployment methodology specifically tailored for mobile platforms that incorporates metrics beyond traditional classifier performance.

✉ Omar Boursalie
boursao@mcmaster.ca

[1] School of Biomedical Engineering, McMaster University, Hamilton, ON, Canada

[2] Department of Computing and Software, McMaster University, Hamilton, ON, Canada

[3] eHealth Graduate Program, McMaster University, Hamilton, ON, Canada

[4] Department of Electrical and Computer Engineering, McMaster University,
Hamilton, ON, Canada

Keywords Remote patient monitoring · Wearable system · SVM · MLP · Data mining · Mobile device · Health records · Severity estimation · Data fusion · Machine learning · System development

1 Introduction

Chronic diseases such as cardiovascular disease (CVD) are an increasing burden for global health-care systems as the population ages [1]. As a result, there is growing interest in developing remote patient monitoring (RPM) systems to assist health professionals in the management of chronic diseases by analyzing immense data collected from wearable sensors and health record data. Generally, the analysis is completed using machine learning algorithms (MLA) [2, 3] resided on remote servers that can handle expensive computational operations. Advances in mobile technology provide new opportunities to deploy MLAs locally on mobile devices lowering transmission expenses and allowing the system to work without any interruption when the network connection is poor or non-existent. However, transferring the data analysis from a remote server to a mobile device introduces its own set of challenges. While there is a wealth of research studies focusing on using machine learning algorithms for remote patient monitoring systems (e.g., CVD [4], respiratory [5], diabetes [6]), the characteristics of the implementation environment (such as required computational power, the network bandwidth and the power consumption to train and/or test the algorithm) and its impact on classification performance is rarely investigated.

In this paper, we systematically study the impacts of the design decisions, made during a mobile RPM system development, on the system's classification and computational performance. We adapt the Yin's case study methodology [7] to investigate the challenges we faced in the design, implementation and deployment of the multi-source mobile analytic RPM system, M4CVD (Mobile Machine Learning Model for Monitoring Cardiovascular Disease) [8]. Four classes of challenges for developing a mobile monitoring system are investigated: data collection, data processing, machine learning and system deployment. We also present our recommendations for addressing the main challenge for each development stage. As part of our recommendations, we propose a novel training and deployment methodology for MLAs on mobile platforms that incorporates additional metrics beyond classification performance.

The paper's contributions and structure are as follows: Section 2 provides an overview of the research model and case study methodology used in this paper. In Section 3, we describe the implementation procedure and challenges we encountered during system development. In Section 4 we present our recommendations for addressing the main challenges identified at each development stage. Section 5 describes the related research. We conclude in Section 6.

2 Research Method

Early remote patient monitoring systems were signal acquisition platforms that continuously transmitted physiological data from a single sensor to a remote server for

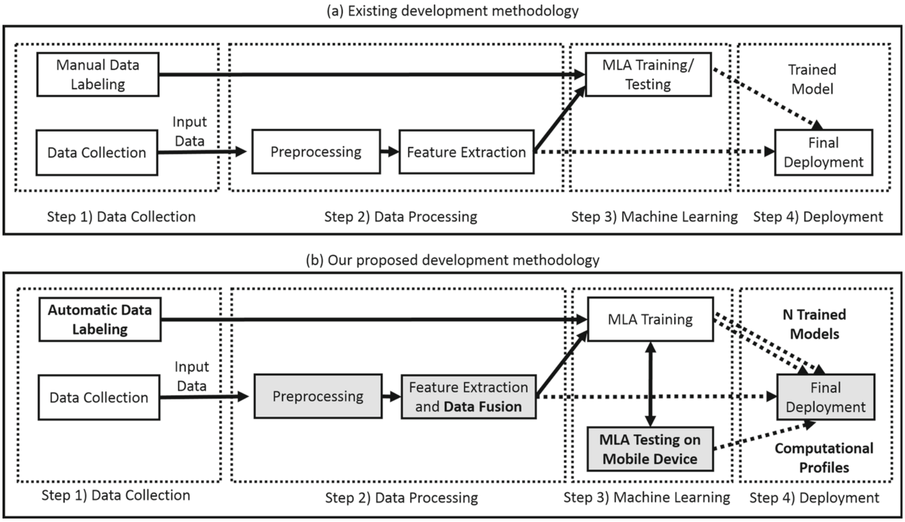


Fig. 1 An overview of **a** current and **b** our proposed methodology for developing a MLA-based remote monitoring system. New components are in bold. Stages in white are on a remote sever while stages in gray are on a mobile device

analysis. Increasingly, monitoring systems are using machine learning algorithms to automatically analyze the collected data which have been shown to increase prediction accuracy with less strict assumptions compared to statistical methods [5]. Regardless of the algorithm, the most common approach used in the development of a machine learning-based monitoring system is shown in Fig. 1a. First, the training data is collected and manually labeled. Next, preprocessing, feature extraction and data fusion techniques are selected to transform the input data into a set of features suitable as inputs to the classifier. Finally, the machine learning algorithm is trained and tested. Most monitoring systems data processing and analysis stages are developed and deployed on remote servers since both stages have a complexity order of approximately $O(n)^3$ [9].

In this research, we investigate how the complexity described above can be managed when the mobile platform is considered as an additional dimension on a remote monitoring system design. We group the challenges we encountered according to Fig. 1a. As part of our methodology we are interested in extending the model described in Fig. 1a to answer the following queries: (1) What are the challenges of monitoring heterogeneous data sources? (2) What are the computational requirements of a monitoring system on a mobile device? (3) How can the computational requirements of a mobile platform be incorporated into the training, testing, and deployment of machine learning algorithms? (4) What are the trade-offs between classifier accuracy and mobile computational performance?

Following the case study methodology [7], we systematically encoded our observation, challenges and design decisions made at every stage of system development shown in Fig. 1a. Our objective was to investigate the main challenges for each development stage. We identified four general decision milestones faced during the development of the mobile-based RPM system with cascading effects on system performance: (1) training data labeling method, (2) data fusion technique, (3) classifier selection, and (4) adapting classifier requirements based on current computational environment. For each milestone a number of alternatives were studied by creating a set of sister RPM systems and evaluating each system in terms of its classification and computational performance.

In Section 3 we discuss the challenges we encountered grouped in terms of the development stages shown in Fig. 1a. We also explore how the design decisions made during system development impacts the model's classification and mobile computational performance. The challenges we identified are solely based on our experience developing M4CVD. However, from related studies we identified that challenges in model training [3, 10] and deployment [11] are generic to developing any MLA-based mobile systems.

Based on our findings, in Section 4 we propose a series of recommendations for addressing the four main decision milestones shown in Fig. 1a. As part of our recommendations, we extend Fig. 1a by proposing a new training and deployment methodology for MLAs on mobile platforms as shown in Fig. 1b. First, we investigate two methods to label training data automatically. Next, we present a comparative analysis of two data fusion techniques for combining heterogeneous data. Third, we propose a novel training methodology for mobile-based MLAs. Currently, classifier training and testing are completed on a remote server. We propose conducting the classifier testing on a mobile device to create accuracy-computational profiles for each candidate model. Our proposed method allows developers to study the tradeoffs between a candidate classifier's accuracy and computational requirements to improve system efficiency. Finally, we propose deploying multiple models with various accuracy-computational profiles to the mobile device. The system can then dynamically select the best model to use based on real-time computational resource availability.

3 RPM Development

In this section, we describe the system development process and identify the challenges we encountered for each stage in Fig. 1a. In Section 3.1 we discuss the data collection stage. Next, we discuss the data processing stage in Section 3.2. Section 3.3 presents a comparative analysis of two machine learning algorithms: 1) Support vector machine (SVM) and 2) Multilayer perceptron (MLP). In Section 3.4 we describe the deployment environment and evaluate the RPM system's mobile computational requirements.

3.1 Data Collection

The first step in data collection is to determine the monitoring system's input sources. Monitoring systems are increasingly analyzing data from a variety of heterogeneous sensors such as ECG and blood pressure (BP) devices to monitor a patient's physiological deterioration [12]; interested readers are referred to [9] for a review

Table 1 Training data baseline characteristics

Clinical features	Labeling technique	Low-risk (LR) class	High-risk (HR) class	*p* value between LR and HR
Age	DRG	68.7 ± 16.0 (n = 267)	67.1 ± 14.0 (n = 251)	0.238
	SAPS I	65.0 ± 15.6 (n = 273)	70.1 ± 15.6 (n = 229)	*0.00001*
Gender (Female)	DRG	40.4% (n = 267)	32.3% (n = 251)	0.053
	SAPS I	34.4 % (n = 229)	38.9% (n = 229)	0.304
Weight (kg)	DRG	81.5 ± 21.5 (n = 267)	85.5 ± 18.3 (n − 251)	*0.021*
	SAPS I	84.4 ± 21.2 (n = 273)	82.8 ± 18.9 (n − 229)	0.36
Systolic blood pressure (mmHg)	DRG	148.5 ± 30.8 (n = 267)	137.2 ± 32.1 (n = 251)	*0.00005*
	SAPS I	144.7 ± 30.0 (n = 273)	141.2 ± 33.9 (n = 229)	0.22
Diastolic blood pressure (mmHg)	DRG	92.2 ± 22.8 (n = 267)	81.7 ± 25.3 (n = 251)	*0.000001*
	SAPS I	89.3 ± 23.1 (n = 273)	84.1 ± 25.8 (n = 229)	*0.02*

DRG diagnosis-related group [15], *SAPS I* simplified acute physiology score I [16]

Italicized *p* values are clinically significant ($\alpha < 0.05$)

on wearable technology. In addition, the growing accessibility of electronic health records using mobile devices [4] provides new opportunities for monitoring systems to analyze sensor physiological data within the context of a patient's clinical data. The next data collection step is to collect the training data. Currently, the data collection step is conducted internally to give researchers full control over their training dataset composition. However, creating a training set containing heterogeneous data suitable for our study is a very challenging and time-consuming task. Instead, we decided to share our experience using the Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II) database [13] to develop our system. We selected the MIMIC-II database because it contains both a physiological and clinical database of anonymized intensive care unit (ICU) patients [13]. Patients with heart disease were identified in the MIMIC-II database as those with a primary International Classification of Diseases (ICD-9) code between $390 - 459$ [14]. In total, 502 heart disease patients with matched physiological and clinical records were identified in the MIMIC-II database. The breakdown of low- and high-risk patients is shown in Table 1. The techniques for labeling the data are presented in Section 4.1. A two-sample *t* test was used to compare continuous variables (e.g., age) while a chi-square test was to compare categorical variables (e.g., gender) between the low and high-risk groups with a *p* value less than $\alpha = 0.05$ deemed significant. Our results show that age, weight, systolic and diastolic blood pressure were different between low and high-risk groups. Subsequently, machine learning algorithms may be able to separate the classes by constructing a hypersurface. Using an open source dataset decreases system development and allows researcher access to larger and more diverse training sets.

Working with wearable sensors, health records, and published datasets not created specifically for our study have its own set of challenges which we summarize

Table 2 Data collection challenges

Data collection stage	Challenges
Wearable sensors	1. Quality of consumer devices remains too low 2. Difficult to get different devices communicating with each other
Health records	1. Data is unstructured 2. Technical, security, and privacy challenges
Training dataset	1. Most researchers currently publish only the final feature set 2. Storage standards are not sufficient for signal processing and machine learning applications 3. Restricts the features that can be studied

in Table 2. First, the quality of both wearable sensors and health records remains a challenge when developing monitoring systems. Despite recent advances, the quality of consumer devices remains too low for medical applications [17]. Similarly, health record data is mostly unstructured and must be converted into a data format suitable for automated analysis [18]. For example, important clinical data (e.g. patient habits) are currently stored as narrative notes that cannot be natively processed by a computer, thus requiring the development of context-specific natural language processing techniques. Next, there is a need to develop communication protocols to allow devices from multiple vendors to communicate with the monitoring system [19]. There are also technical, security, and privacy challenges for integrating an external monitoring platform with a hospital health record system. Third, most researchers currently only publish their studies final feature set (e.g., UCI [20]) which have limited use outside the scope of the original study. In addition, the data quality of online repositories may prevent the analysis of the data using machine learning or signal processing. For example, Physionet's current guidelines [21] only set the minimum requirements to ensure a physiological dataset's compatibility with the waveform viewers which is not suitable for all research applications. There is a need to develop standards for online repositories to enable future signal processing and machine learning applications. Overall, the biggest challenge in the data collection stage was labeling the training examples so they can be analyzed using supervised machine learning algorithms. Labeling the training set (e.g., low or high disease severity) is usually completed manually by a medical expert which can be very time consuming [22] and limits the size of the training set. In Section 4.1, we investigate two methods for automatically labeling severity of a patient being at risk of cardiovascular disease.

3.2 Data Processing

The heterogeneous data must be processed into a set of features suitable for analysis using MLAs. Our data processing stage consists of: 1) Wearable sensor preprocessing, 2) Health record imputation, and 3) Feature extraction. First, sensor preprocessing is used to improve the quality of physiological signals which suffer

Table 3 The 11 features from ECG and BP sensors and health records monitored by M4CVD. *C* continuous features, *D* discrete feature

Clinical data	Blood pressure sensor	ECG sensor
1. Gender (D)	4. Systolic blood pressure (D)	6. Heart Rate (D)
2. Age (C)	5. Diastolic blood pressure (D)	7. Mean R-R interval (C)
3. BMI (D)		8. Heart rate variability (C)
		9. Standard deviation of R-R (SDNN) (C)
		10. Square root of mean difference of R-R (rMSSD) (C)
		11. Percentage of R-R interval greater than 50 ms (pNN50) (C)

from noise and motion artifact [23]. Specifically, the ECG signal undergoes four preprocessing steps: filtering [24], detrending, ECG signal quality assessment [25] and R peak detection [26]. Next, imputation methods are used to deal with the missing and incomplete data in health records [27]. For example, in our training database 33% of health records were missing data on patient height. We used regression imputation where patients with known age, weight, and height [28] were used to construct a 2^{nd} order height imputation model. The final data processing stage is feature extraction for converting continuous physiological signals into discrete values. Our feature extraction stage primarily focused on extracting time, heart rate variability [29], and frequency features [30] from 5 minute ECG signals in the MIMIC-II physiological database. No additional feature extraction for BP recordings and health records was necessary because they already contain the features of interest. After reviewing the literature, we identified twenty-four prospective features extracted from ECG, BP, and health records that are used for monitoring CVD. Eleven features (Table 3) were successfully implemented and validated for further study.

The process for selecting the final feature set is rarely discussed in the literature beyond the use of feature selection algorithms [31]. However, in our experience the primary feature selection criteria is not a feature's contribution to model accuracy but rather identifying features that can be successfully extracted and validated. While the data processing challenges summarized in Table 4 are context-specific it is important to discuss these challenges to serve as a guide for future developers of data processing libraries and monitoring systems. First, proposed ECG preprocessing libraries are mostly tested on gold standard datasets [32] which have less noise and motion artifacts compared to wearable sensor data. There is a need for a gold standard database of ECG recordings from wearable sensors. Second, selecting the proper health record imputation method is a challenge because each method introduces their own level of uncertainty [33]. Third, the ECG recordings in the MIMIC-II database underwent signal decimation destroying the ECG signal's frequency component. As a result, both the ECG detection libraries [21] and the frequency domain feature were not successfully validated. It is outside the scope of this paper to improve the automatic peak detection methods. Only features extracted from the R peak (heart rate, R-R interval

Table 4 Data processing challenges

Data processing stage	Challenges
Wearable sensor preprocessing	1. Existing libraries have been validated using gold standard waveform datasets not data collected from wearable sensors
Health record preprocessing	1. Selecting the proper imputation method depends on the type of health data being studied
Feature extraction	1. Few feature extractions were successfully validated due to the poor quality of the training data 2. No guidelines for selecting the data fusion technique for combining heterogeneous data

heart rate variability, SDNN, rMSSD and pNN50) [29] were included in the final feature set. The training dataset also rarely included information on patient habits (e.g. smoking and exercise) which were excluded from study. Finally, a common challenge working with heterogeneous data is selecting the data fusion technique to combine the data for analysis using MLAs. In Section 4.2 we present a comparative analysis of two data fusion techniques for combining data from wearable sensor and health records.

3.3 Machine Learning

The third step as shown in Fig. 1a was the design and training of the SVM and MLP to predict low or high disease severity. Both classification algorithms are popular in the medical domain [23] due to their ability to map features to higher dimensional space: the SVM using kernel functions while the MLP uses hidden layers [9]. Interested readers are referred to [34] and [35] for a detailed explanation on the SVM and MLP respectively. Both classifiers were trained and tested on the dataset of 502 patient records containing 11 features extracted from wearable sensors and health records. The LibSVM machine learning library [36] and MATLAB's neural network toolbox was used to implement the SVM and MLP respectively. The SVM was trained using 10-fold cross-validation (CV) training with 70% of the dataset for training and 30% for testing. For MLP training the dataset was divided into 80% training and 20% testing sets with 25% of the training data used as the validation set (The cross-validation results are presented in Section 4.3). Then, the best SVM and MLP configurations were tested using a Monte Carlo simulation where each algorithm was trained and tested 1000 times on a random subset of training examples. No patient record was used in both the training and testing set during the same simulation run. The Monte Carlo results and mean receiver-operator curves (ROC) [37] for each classifier are shown in Table 5 and Fig. 2 respectively. Both models achieved stable and reusable parameter configurations. Our results show that the SVM had the best overall performance. The SVM appears to generalize consistently across simulation runs as the

Table 5 M4CVD Performance for SVM and MLP. The mean of 1000 experiments is shown

Classifier	Accuracy		
	Max	Min	Mean
SVM	71.30%	49.00%	62.5 ± 3.64 %
MLP	77.50%	42.20%	60.7 ± 5.99 %
Classifier	Area under the ROC	Sensitivity	Specificity
SVM	66.00 ± 3.00%	45.53 ± 7.04%	76.21 ± 6.01%
MLP	65.19 ± 7.53%	60.37 ± 8.18%	61.03 ± 7.88%

SVM always finds the global minima solution. On the other hand, the MLP update it's weights and bias individually so it is more sensitive to the variability within each feature. Based on classifier accuracy we would recommend the SVM for CVD severity classification. The best SVM and MLP were then deployed to a mobile environment for further testing as discussed in Section 3.4.

Our results are promising since they do exceed those of current early-warning system which monitor physiological indicators [38]. The early-warning system was implemented in twelve hospitals over a six month period and identified 30% (95/611) of patients who were subsequently admitted to the ICU. Nevertheless, existing algorithms are designed for analyzing homogeneous data from a single data source. As

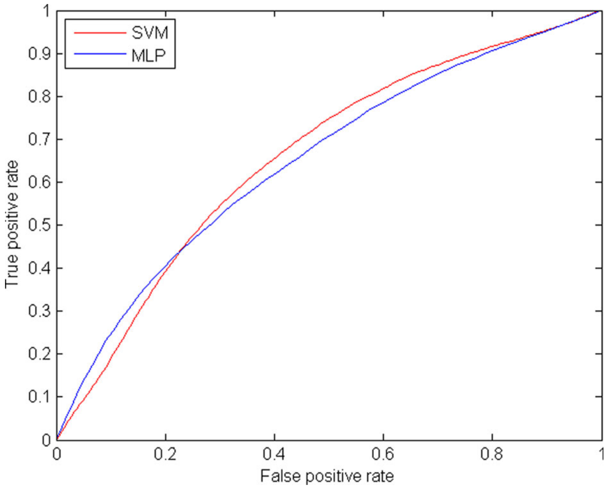


Fig. 2 ROC curve for severity estimation. The mean of 1000 experiments have been shown for each classifier

Table 6 Machine learning challenges

	Challenges
Model inputs	1. Current machine learning models are designed for homogeneous datasets
	2. MLA need to deal with structured, semi-structured, and unstructured data simultaneously
Training	1. MLA cannot incorporate new data once deployed without retraining the whole system under constant expert supervision
	2. Cross-validation training only maximizes classifier accuracy
	3. The classifier's precision and recall or F1 scores are not always presented when reporting model performance

a result, there a number of challenges (Table 6) using machine learning for analyzing heterogeneous data on a mobile device. First, there is a need for new algorithms that can analyze heterogeneous datasets [39]. Such algorithms will need to deal with structured, semi-structured, and unstructured data simultaneously [40]. Second, deployed MLAs cannot incorporate new data without expert supervision. Third, the main challenge we identified is that the current classifier training methodology focuses on determining the model configurations that maximizes the model's classification performance (e.g., accuracy). In Section 4.3 we propose a new training methodology for machine learning that evaluates a model using classification performance and mobile computational complexity. Finally, many RPM systems we reviewed (Section 5) were only evaluated using accuracy which can lead to suboptimal solutions [41]. Researchers should also report precision and recall or F1 scores when discussing classifier performance.

3.4 Deployment and Hardware Evaluation

In this paper, the development and deployment of M4CVD was done on different target hardware. We used a 64-bit Windows 7 laptop with a 2.2 GHz Intel i7 CPU and 12 GB RAM using MATLAB 2014A for developing the monitoring system. The final system was then deployed in C++ to a Linux Raspberry Pi 2 Model B (RASPI), a single board computer (Quad-core, ARMv7, 900 MHz CPU, 1 GB RAM) with similar performance to the low-cost 2014 Motorola Moto G.

Table 7 shows the computational requirements for the input, data processing, and deployed classifier modules. Our initial hypothesis was that machine learning models present a considerable burden for low resource devices because they have a complexity order of approximately $O(n)^3$ [9]. Surprisingly, our results show that the analysis stage required among the lowest computational resources in terms of execution time and current consumption. Instead, the signal acquisition and data processing modules were major computational bottlenecks in our mobile monitoring system. The most computationally expensive components in our system were the ECG quality assessment and R peak detection stages due to a large amount of raw physiological data

Table 7 Hardware consumption for acquisition, data processing and deployed classifier modules on Raspberry Pi 2

	Input	Data processing	Deployed MLA	
			SVM	MLP
CPU usage	1.39 ± 0.01%	26.434 ± 6.63%	38.28 ± 6.11%	*54.03 ± 16.44%*
Memory usage	0.20%	*0.606% ± 0.44%*	0.12%	0.10%
Execution time	322. 5 sec (5 min)	*1228.43 ± 152.88 ms*	71.0 ± 1.72 ms	2.65 ± 0.1 ms
Current consumption	*274 ± 6.6 mA*	43.18 ± 11.53 mA	11.7 ± 10.7 mA	6.7 ± 7.5 mA

The highest value for each metric is in italics

processed. Interestingly, Table 7 shows that the support vector machine and multilayer perceptron had very different computational requirements despite their similar classification performances. The SVM took 70x longer and required 2X the current compared to the multilayer perceptron. The different computational requirement appears to be a result of how each model classifies new data after deployment. The SVM constantly maps each input data vector into higher dimensional space using the kernel function which can be computationally expensive. On the other hand, once deployed the MLP is a series of equations requiring less computational resources. Overall, our results demonstrate that the MLA's complexity was not a barrier for adoption on a mobile device. In fact, our findings suggest that many RPM systems already run the most computationally expensive modules (data collection and processing) locally. We recommend the MLP for deployment in a mobile monitoring system because the MLP has similar classifier performance and superior mobile computational performance compared to the support vector machine.

Deploying a monitoring system to a mobile device and evaluating the system's computational performance is a non-trivial and time-consuming task (Table 8). First, there is a need for preprocessing and machine learning libraries that are optimized for deployment on a mobile device. For example, the support vector machine can be implemented using fixed-point arithmetic which is less computationally expensive [42]. Next, developers should consider both accuracy and computational power when selecting the preprocessing techniques, features, and classifier for their monitoring systems. Third, popular MLA libraries [43, 44] assume that model training and deployment occurs in the same computational environment. Future libraries should support training and deployment to different platforms natively. The next generation of RPM systems will be deployed entirely on mobile devices with little communication with remote servers. However, the main challenge with existing mobile RPM systems such as M4CVD is that they have constant computational requirements regardless of the current usage environment. In Section 4.4, we propose a methodology to allow a monitoring system to adapt their classification module based on user preferences and the current system condition. However, evaluating the computational requirements for mobile systems requires its own experimental procedure and setup which extends classifier training and system development time.

Table 8 Deployment and mobile computational requirement challenges

Deployment stage	Challenges
Data processing	1. Current preprocessing libraries are not computationally efficient 2. Feature selection techniques only consider each feature's contribution to accuracy
Machine learning	1. MLA libraries assume that model training and deployment occurs in the same computational environment 2. Both classification performance and computational power should be considered when evaluating classifiers for mobile systems
Entire system	1. Evaluating the computational requirements of a monitoring system requires its own experimental procedure 2. Computational requirements for each stage remains constant once deployed and cannot adapt to the current usage environment

4 Recommendations

In this section we propose a system development methodology (Fig. 1b) that addresses the four main decision points identified in this paper: 1) training data labeling method, 2) heterogeneous data fusion, 3) optimizing machine learning classifiers for a mobile environment, and 4) adapting MLA based on current computational requirements. In Section 4.1 we investigate using automatic techniques to label our training set. Section 4.2 compares two data fusion techniques (feature and decision-level fusion) for combining heterogeneous data sources. Note that our recommendations for automatic data labeling and heterogeneous data fusion are based on our experience developing M4CVD and are domain specific. We also propose a machine learning training methodology that considers both classification performance and computational cost during cross-validated training in Section 4.3. In Section 4.4 we propose a deployment methodology for dynamically selecting the best classifier based on the current computational resources available on a mobile device. Our recommendations for extending the MLA training and deployment methodology can be used when developing classifiers for any mobile application.

4.1 Data Collection

In this section, we investigate two methods to automatically label the disease severity of the 502 patient records used to train M4CVD: 1) Simplified Acute Physiology Score I (SAPS) [16] and 2) Diagnosis Related Group (DRG) [15]. SAPS is an intensive care unit (ICU) patient severity scoring system. DRG is a USA hospital payment classification system that measures the relative amount of resources used to treat the patient which we use as an indicator for patient severity. Both metrics are calculated by health professionals during the patient's hospital stay and stored in the MIMIC-II database.

Once the SAPS and DRG scores were retrieved for each patient record, the next step was to separate the training examples into low and high severity classes using the automatic prioritization of ICU patients method proposed by [45, 46]. High-risk patients were defined as those whose severity score was above the calculated median scores. Overall, 54 and 51% of patient examples were labeled high severity based on their SAPS and DRG score respectively. Table 9 compares the classification results for each labeling technique across a subset of the classifier configurations tested. Our results show that both models could be trained to distinguish between low and high-risk patients using data labeled automatically by the SAPS or DRG metrics. The support vector machine had higher classification performance using the SAPS while the multilayer perception showed improved performance using the DRG labels.

Automatic labeling offers several advantages. First, automated labeling enables developers to build models using larger datasets compared to datasets that are labeled manually. Next, automatic labeling is a method for incorporating pre-existing medical knowledge into MLAs. Third, automatic labeling reduces system development time. Automatic labeling can serve as a preprocessing step to evaluate the distribution of a dataset and identify the best data subset for manual expert labeling. However, automatic labeling can be domain specific and time-consuming to develop. In addition, an important area to investigate is the agreement between labels generated by automated techniques and human experts. Finally, automatic labeling may not always be available. An alternative labeling method is unsupervised learning [47] which is a class of algorithms used to discover hidden patterns or groupings from unlabeled datasets. Interested readers are referred to [48] for a detailed explanation on unsupervised learning.

4.2 Data Processing

A data fusion stage is increasingly used in monitoring systems to combine heterogeneous data into a single higher dimension feature vector. Multiple data fusion techniques have been used in the literature; interested readers are referred to [49] for a full review. However, as far as we know a comparison between fusion methods on the same monitoring system has not been presented. In this section we compared two data fusion techniques on a mobile device: (1) feature-level and (2) decision-level fusion [50, 51]. While our comparison in this section is domain specific, our recommendations can serve as a starting point for researchers developing systems that combine data from heterogeneous sources.

Feature-level fusion is the simple concatenation of heterogeneous features into a single input vector [52]. However, each extracted feature has their own numeric ranges which present a challenge. During training features with large physiological ranges may be assigned more weight regardless of the importance of the feature to classification accuracy [53]. The range bias can be removed by normalizing all features to a range of (0,1). Feature-level fusion can be very powerful because it allows us to correlate features across data sources and is not computationally expensive. However, feature-level fusion requires a large training dataset in order to apply feature selection algorithms [31]. On the other hand, decision-level fusion allows us to incorporate medical knowledge directly into our model. Before concatenation, each

Table 9 Comparison of SAPS I and DRG automatic labeling techniques for cross-validation (k = 10) training

Classifier	Kernel/Learning function	Labeling technique	Accuracy (%)	AUC (%)	Precision (%)	Recall (%)	F1 (%)
SVM	Radial basis function	SAPS I	*64.53 ± 8.96*	67.37 ± 9.89	63.55 ± 12.92	*56.80 ± 7.34*	*59.63 ± 8.57*
		DRG	63.12 ± 6.72	*68.06 ± 5.87*	*63.96 ± 8.21*	56.22 ± 9.89	59.47 ± 7.54
	Polynomial (5th order)	SAPS I	*61.95 ± 5.05*	61.67 ± 6.37	*60.57 ± 6.53*	48.93 ± 11.48	53.43 ± 8.35
		DRG	60.44 ± 7.05	*63.40 ± 8.01*	57.14 ± 5.57	*71.71± 11.39*	*63.48 ± 7.66*
	Sigmoid	SAPS I	*64.93 ± 7.94*	*67.77 ± 9.94*	62.80 ± 10.36	58.44 ± 10.14	60.24 ± 9.17
		DRG	62.92 ± 5.88	66.83 ± 5.69	*63.05 ± 9.65*	*60.60 ± 9.24*	*61.15 ± 6.31*
MLP	Levenberg-Marquardt	SAPS I	63.80 ± 4.85	62.34 ± 5.12	*67.90 ± 15.56*	43.52 ± 7.62	51.83 ± 6.71
	Backpropagation	DRG	*64.51 ± 7.70*	*62.55 ± 7.13*	57.85 ± 23.77	*58.00 ± 25.11*	*56.83 ± 22.40*
	Scaled conjugate gradient descent	SAPS I	65.00 ± 8.71	64.08 ± 8.75	63.23 ± 10.84	54.00 ± 11.96	57.97 ± 10..66
		DRG	*65.10 ± 8.05*	*65.51 ± 8.16*	*65.29 ± 11.95*	*62.38 ± 11.07*	*63.04 ± 8.51*
	Gradient descent	SAPS I	63.00 ± 8.07	61.43 ± 8.13	61.22 ± 12.73	47.97 ± 12.11	53.50 ± 11.95
		DRG	*63.33 ± 8.42*	*63.41 ± 8.21*	*62.74 ± 8.85*	*59.30 ± 13.31*	*60.32 ± 9.59*

The highest value for each metric is in italics

Table 10 Decision-level data fusion local decision ranges for each feature

Feature	Wearable sensor?	Clinical database?	Physiological range	Decision-level fusion format
Body mass index	N	Y	Normal < 24 kg/m^2	1
			Overweight $25 - 29.9$ kg/m^2	2
			Obese I $30 - 39.9$ kg/m^2	3
			Obese II > 40 kg/m^2	4
Systolic blood	Y	Y	Low risk < 120 mmHg	0
Pressure			Medium Risk $121 - 139$ mmHg	1
			High risk > 140 mmHg	2
Diastolic blood	Y	Y	Low risk < 80 mmHg	0
Pressure			Medium risk $80 - 89$ mmHg	1
			High risk $>$90 mmHg	2
Heart rate	Y	Y	Normal $60 - 100$ beats/min	0
			Abnormal other	1
R-R interval	Y	N	Normal $0.4 - 1.5$ s	0
			Abnormal $>$1.5s	1

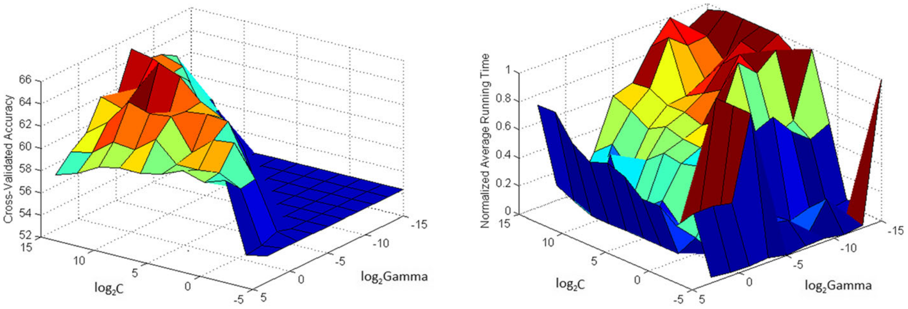
feature is first evaluated individually to make a local decision. The classifier then makes a high-level decision by analyzing all the local decisions [52]. In this paper, healthy and unhealthy ranges set by The Canadian Heart and Stroke Foundation [54] were used for each local decision (Table 10). Each feature was assigned a category corresponding to each its range of healthy and unhealthy values (e.g., 1–4) and normalized to remove range bias. Features without healthy and unhealthy ranges (e.g., age) were normalized.

Both feature and decision-level fusion were tested across all classifier training configurations, a subset of results is shown in Table 11. Interestingly, both models showed improved performance using feature-level fusion that did not incorporate any *a priori* medical knowledge. Our results demonstrate the risk of injecting designers' bias into the model using decision-level fusion. For example, the physiological ranges used in Table 10 are based on the overall healthy population. However, our training set on average has higher mean values for each feature compared to the overall population because the patients have CVD. As a result, the local decision assigns many of the training patient's features as medium or high risk (few low risk) reducing the classifier's sensitivity. On the other hand, when no decision-level fusion is conducted the machine learning algorithm determines for itself the relative importance of each input feature individually without the need for expert input. The MLP considers each feature importance by updating each weight and bias individually through back-propagation [55]. Both feature and decision-level fusion were not computationally expensive but decision-level fusion does introduce additional computational overhead.

Table 11 Comparison of feature and decision-level data fusion techniques for cross-validation (k = 10) training

Classifier	Kernel/Learning function	Data fusion mode	Accuracy (%)	AUC (%)	Precision (%)	Recall (%)	F1 (%)
SVM	Radial basis function	Feature	64.53 ± 8.96	*67.37± 9.89*	63.55 ± 12.92	*56.80 ± 7.34*	59.63 ± 8.57
		Decision	*65.94 ± 9.27*	67.12 ± 10.46	*64.52 ± 11.19*	56.78 ± 11.62	*60.18 ± 10.92*
	Polynomial (5th Order)	Feature	*61.95 ± 5.05*	*61.67 ± 6.37*	*60.57 ± 6.53*	48.93 ± 11.48	53.43 ± 8.35
		Decision	59.93 ± 5.73	59.91 ± 7.09	57.15 ± 6.81	*51.50 ± 6.82*	*53.95 ± 5.80*
	Sigmoid	Feature	*64.93 ± 7.94*	*67.77 ± 9.94*	*62.80 ± 10.36*	*58.44 ± 10.14*	*60.24 ± 9.17*
		Decision	63.60 ± 6.67	66.39 ± 9.44	61.49 ± 9.20	56.80 ± 8.15	58.72 ± 7.24
MLP	Levenberg-Marquardt	Feature	*63.80 ± 4.85*	62.34 ± 5.12	*67.90 ± 15.56*	43.52 ± 7.62	51.85 ± 6.71
	Backpropagation	Decision	63.00 ± 7.07	*62.71 ± 7.27*	59.25 ± 9.24	*61.19 ± 18.70*	*58.76 ± 11.57*
	Scaled conjugate gradient descent	Feature	*65.00 ± 8.71*	*64.08 ± 8.75*	63.23 ± 10.84	*54.00 ± 11.96*	*57.97 ± 10.66*
		Decision	63.20 ± 7.38	62.58 ± 7.30	*64.78 ± 13.22*	49.10 ± 15.99	53.68 ± 10.54
	Gradient descent	Feature	*63.00 ± 8.07*	*61.43 ± 8.13*	*61.22 ± 12.73*	*47.97 ± 12.11*	*53.50 ± 11.95*
		Decision	61.29 ± 4.34	60.36 ± 4.28	59.98 ± 8.55	46.55 ± 10.20	51.67 ± 8.32

The highest value for each metric is in italics



Highest Accuracy	Running Time
65.33%	1.1 ms
64.94%	1.0 ms
64.56%	1.1 ms

(a)

Shortest Running Time	Accuracy
0.77 ms	59.37%
0.82 ms	58.34%
1.1 ms	60.76%

(b)

Fig. 3 The proposed cross-validation procedure examines both accuracy (**a**) and normalized execution time (**b**) to identify the best overall SVM classifier

4.3 Machine Learning

Currently, the objective of training MLAs is to determine the best architecture (e.g., kernel and learning function) and user-defined parameters (e.g., C, gamma, number of neurons) that maximize the model's classification performance. Our proposed methodology extends MLA training to evaluate each model configuration's classification performance (e.g., accuracy) and mobile computational requirements. First, each configuration is trained and tested using the traditional cross-validation technique. For example, Fig. 3a shows the traditional cross-validation accuracy results for the SVM presented in Section 3.3. Next, each model is deployed to the target mobile device and evaluated in terms of current consumption, execution time, CPU and memory usage. As a work in progress, we evaluated the SVM training and computational testing on a Windows 7 laptop with 2.2 GHz Intel i7 CPU and 12 GB RAM using MATLAB 2014A. Finally, a cross-validation graph showing how the performance metrics change with different model configurations was generated. Figure 3b demonstrates how the SVM's configuration effects the model's execution time. Developers can use Fig. 3 to study the trade-offs between a classifier's accuracy and efficiency. For example, examining Fig. 3 the highest classifier accuracy was 65.3% and took 1.1 ms to run. However, the developer may decide that a 5% decrease in accuracy (65.3% down to 60%) is an acceptable trade-off to save 36% in execution time (1.1 ms down to 0.7 ms) increasing the monitoring systems' operation time. The optimal model is now the one that balances both accuracy and execution time.

Our proposed training methodology provides developers a better indicator of their classifiers overall performance. The proposed methodology can be used when developing classifiers for any mobile application as our method extends the MLA training procedure. However, our methodology will increase the model's training time compared to traditional cross-validation training since every candidate model is deployed

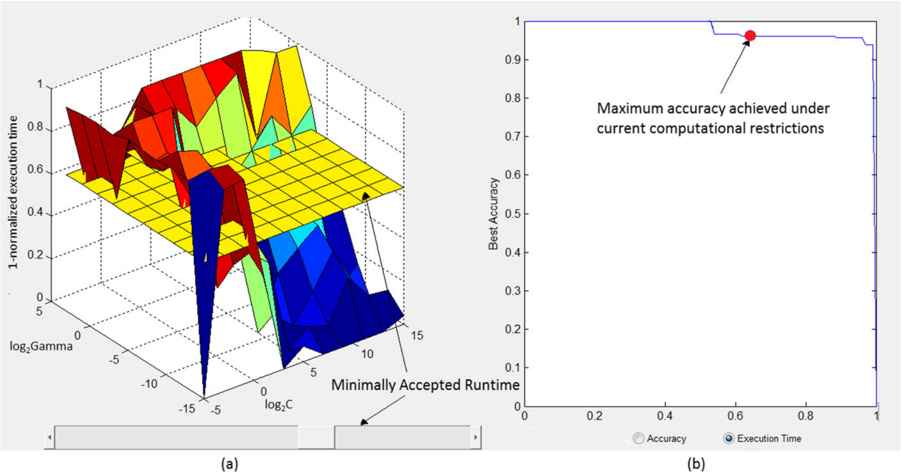


Fig. 4 The proposed deployment model allows the user to select the trade-off between the SVM's computational usage (**a**) and accuracy (**b**)

and tested on the mobile device. In addition, our proposed methodology would require the development of an automated procedure to deploy the classifier to the mobile device and evaluate its computational performance.

4.4 Deployment

The final stage in Fig. 1b is deploying the classifier to the mobile device. However, once deployed existing monitoring systems cannot adapt their model's computational resources based on real-time resource availability. A potential solution is to deploy multiple classifiers with various accuracy-computational profiles to the mobile device. Our study shows that multiple classifiers can be stored on a mobile device due to each model's small storage requirement (SVM- 68 KB, MLP- 20 KB). Figure 4 shows our proposed model for selecting the best classifier. Figure 4a shows the normalized run times for 100 SVMs. In this paper, we assume the model with the shortest run time also has the lowest resource requirements. The user selects the minimally acceptable runtime they will accept (yellow plane) and Fig. 4b shows the maximum normalized accuracy the system can achieve under the user constraints. In this case, our model shows that there is no trade-off between accuracy and execution time until 0.5 normalized run time after which decreasing the classifier's execution time reduces its classification accuracy. Interestingly, we only need to deploy three of the 100 SVMs to capture the full range of accuracy and computational trade-offs corresponding to the main inflection points in Fig. 4b. Our proposed model will further increase the efficiency of classifiers running on mobile and low resource devices.

The proposed methodology for dynamically selecting a MLA's configuration can be used when deploying any classifier into a mobile environment. In addition, the proposed methodology can be automated as many mobile systems provide access to the device's current computational status (e.g., CPU, RAM, battery life). For example, if the monitoring system's battery life goes below 10% the system can automatically switch to the most efficient classifier to extend the system's operation time. The proposed model allows the user to visualize the trade-off between system accuracy and execution time.

5 Related Work

In this section, we review existing remote monitoring systems in terms of their data collection (Section 5.1), processing (Section 5.2) and analysis (Section 5.3) modules.

5.1 Data Collection

Early RPM proposals measured only a single physiological signal, primarily ECG [56] and activity level [17, 42]. Increasingly, RPM systems are monitoring multiple physiological signals using wearable devices [6, 57] or ICU monitors [58, 59]. However, most monitoring systems we reviewed used the local device for signal acquisition only, despite mobile phones having the computational power to support MLAs [60]. Existing systems also do not integrate with electronic health record repositories despite their growing accessibility on mobile devices [4]. Instead, existing systems only collect and display basic clinical data to the health professionals [61]. In addition, the majority of the papers we reviewed [59, 62–64] had their own internal data collection stage or used an open-sourced database [65]. However, most training records are annotated manually by experts [3, 63, 65, 66]. As a result, the size of training sets in existing studies has been small ranging in size from only a few dozen [62, 63] to a few hundred [59, 64] patients. Existing studies on monitoring systems have focused on describing each system's implementation and accuracy. In this paper, we explored the challenges of developing the acquisition, processing and analysis stages for a monitoring system that analyzed heterogeneous data on a mobile device. We also investigated the use of hospital severity metrics to label a large training set automatically.

5.2 Data Processing

The data processing stage consists of preprocessing, feature extraction and data fusion to combine heterogeneous data. Most physiological preprocessing modules involve low/high pass filtering [56, 67], signal amplification [68] and basic feature detection (e.g., R peak [67]). The features extracted from the preprocessed signals have varied considerably between RPM systems [23] depending on the combination of features that best maximize each system's accuracy. In current systems, the feature extraction stage has occurred primarily on remote servers [2, 3, 65, 69] but is increasingly being completed on low resource devices [10, 60]. While developing preprocessing and feature extraction techniques remains an active area of research [25, 69], the computational requirements for these stages on low resource devices have not been investigated in depth [23]. In this paper, we evaluated the computational requirements for M4CVD's preprocessing and feature extraction stage. Surprisingly, our results show that the preprocessing stage was the most computationally demanding component of our system.

Multi-sensor monitoring systems have traditionally analyzed [2] and displayed [70] each sensor stream independently. Recently, RPM systems have begun to use data fusion techniques to combine data from multiple sources for analysis [49]. Feature-level fusion is the most common data fusion technique used in monitoring systems [2, 59, 71]. Decision-level fusion has also been used to detect abnormal physiological signals [64] and label sensor data with the patient's current activity level [63]. However, existing surveys on sensor fusion techniques [49] do not compare the effectiveness of different techniques using the same RPM system. In this paper, we compared the classification performance and computational requirements for both feature and decision-level fusion in the same monitoring system.

5.3 Machine Learning

Machine learning algorithms are increasingly being used in the medical field for screening, diagnosis, treatment, prognosis, monitoring and disease management [72]. In monitoring systems MLAs are primarily used for novelty detection [2, 69] and severity classification [3, 64, 65] applications. The main limitation of these systems is that the data analysis occurs on remote servers requiring continuous data transmission. Increasing mobile computational power provides new opportunities to deploy MLAs directly on the low resource device. For example, HeartToGo [60] used MLAs deployed on a mobile device to classify ECG signals with an accuracy of 90%. However, HeartToGo only monitors a single wearable sensor. Another example is the CHRONIOUS platform [10], a mobile RPM system for patients suffering from chronic obstructive pulmonary and kidney disease which achieves an accuracy of 95% [10, 73].

Multiple studies have conducted a comparative analysis of MLAs [3, 5]. Overall, the SVM has slightly better performance compared to the MLP in monitoring patient severity. For example, Clifton et al. [2] used ICU monitors to analyze patient respiratory rate, HR, and BP to detect periods of signal abnormality. The SVM performed best out of the five classifiers tested with an accuracy of 95%. Another comparative analysis was conducted during the development of the CHRONIOUS system [10] where both the SVM and MLP achieved a similar accuracy of 89% and 87.5% respectively. Existing comparative analyses have focused on evaluating a system's classification accuracy. However, a key difference between mobile and remote server-based systems is the limited computational resources available on the mobile device. Understanding the system's resource requirements is a key metric to assess the systems overall usability and to identify areas of improvement. Despite this importance, only a few studies have investigated their system's resource requirements in-depth [11, 68, 74]. In this paper, we have evaluated the SVM and MLP in terms of both their classification performance and execution time. We have also proposed a novel training and deployment methodology for MLAs operating on mobile devices.

6 Conclusion, Limitations, and Future Work

Advances in mobile technology provide new opportunities to analyze collected data directly on low and even ultra-low resource devices. However, our findings show that there are specific challenges when monitoring systems are being developed for mobile platforms. In this paper, we presented a case study to systematically investigate the challenges we faced in the design, implementation, and deployment of a mobile monitoring system. Based on our findings, we developed recommendations for each development stage which can be used as guidelines by future researchers, system designers, and developers working on mobile-based monitoring systems. While most of our recommendations are stage-specific, our proposal to evaluate classifiers based on accuracy and computational performance is applicable throughout the development process. For example, MLA features could be evaluated based on their contribution to both model accuracy and computational overhead. The work presented in this paper contributes towards the goal of personalized predictive monitoring.

Our study also exhibits some limitations. First, our recommendations are domain specific and do not account for the data collection, processing and analysis techniques used for monitoring other chronic diseases such as respiratory disease and diabetes. In addition, the implementation challenges for the communication, security and privacy modules for a monitoring system on a mobile device were not investigated in this paper.

In view of these results, our next step is to generalize our methodology by investigating other MLA-based mobile systems. Future work will also focus on developing feature selection and training methodologies that consider both classifier accuracy and mobile computational requirements during the optimization of machine learning algorithms. The training methodology will require heuristic algorithms to automatically find satisfactory solutions in the model configuration search space. We are also investigating MLAs that can incorporate new data without constant expert supervision. Finally, we will consider testing the monitoring system using other classifiers such as random forest trees and multi-class MLAs.

Acknowledgments Support from the McMaster School of Biomedical Engineering, McMaster Science & Research Board (SERB), Vector Institute for Artificial Intelligence, and Natural Sciences & Engineering Research Council of Canada (NSERC) is acknowledged.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. World Health Organization (2010) Chronic disease prevention and health promotion

2. Clifton L, Clifton DA, Watkinson PJ, Tarassenko L (2011) Identification of patient deterioration in vital-sign data using one-class SVMs. In: 2011 federated conference on computer science and information systems (FedCSIS), pp 125–131. https://doi.org/10.1109/icma.2007.4303943
3. Melillo P, Izzo R, Orrico A, Scala P, Attanasio M, Mirra M, De Luca N, Pecchia L (2015) Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis. PloS One 10(3):1–14. https://doi.org/10.1371/journal.pone.0118504
4. Jung EY, Kim J, Chung KY, Park DK (2014) Mobile healthcare application with EMR interoperability for diabetes patients. Clust Comput 17(3):871–880. https://doi.org/10.1007/s10586-013-0315-2
5. Luo G, Stone BL, Fassl B, Maloney CG, Gesteland PH, Yerram SR, Nkoy FL (2015) Predicting asthma control deterioration in children. BMC Med Inform Decis Mak 15(1):84–92. https://doi.org/10.1186/s12911-015-0208-9
6. Katsaras T, Milsis A, Rizikari M, Saoulis N, Varoutaki E, Vontetsianos A (2011) The use of the Healthwear wearable system in chronic patients' early hospital discharge: Control randomized clinical trial. In: 5th international symposium on medical information & communication technology (ISMICT), pp 143–146. https://doi.org/10.1109/ismict.2011.5759815
7. Yin RK (2013) Case study research: design and methods. Sage Publications, Thousand Oaks
8. Boursalie O, Samavi R, Doyle T (2015) M4CVD: Mobile machine learning model for monitoring cardiovascular disease. In: The 5th international conference on current & future trends of information & communication technologies in healthcare (ICTH '15), pp 384–391. https://doi.org/10.1016/j.procs.2015.08.357
9. Andreu-Perez J, Leff DR, Ip H, Yang GZ (2015) From wearable sensors to smart implants—Toward pervasive and personalized healthcare. IEEE Trans Biomed Eng 62(12):2750–2762. https://doi.org/10.1109/TBME.2015.2422751
10. Bellos C, Papadopoulos A, Rosso R, Fotiadis DI (2011) Heterogeneous data fusion and intelligent techniques embedded in a mobile application for real-time chronic disease management. In: Annual international conference of the ieee engineering in medicine and biology society (EMBC '11), pp 8303–8306. https://doi.org/10.1109/IEMBS.2011.6092047
11. Comito C, Talia D (2015) Evaluating and predicting energy consumption of data mining algorithms on mobile devices. In: IEEE international conference on data science and advanced analytics (DSAA '15), pp 1–8. https://doi.org/10.1109/DSAA.2015.7344848
12. Buist MD, Jarmolowski E, Burton PR, Bernard SA, Waxman BP, Anderson J (1999) Recognising clinical instability in hospital patients before cardiac arrest or unplanned admission to intensive care. a pilot study in a tertiary-care hospital. Med J Aust 171(1):22–25
13. Saeed M, Villarroel M, Reisner AT, Clifford G, Lehman LW, Moody G, Heldt T, Kyaw TH, Moody B, Mark RG (2011) Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database. Crit Care Med 39(5):952. https://doi.org/10.1097/CCM.0b013e31820a92c6
14. World Health Organization (2010) Burden: mortality, morbidity and risk factors. Global Status Report on Noncommunicable Diseases
15. Averill RF, Goldfield N, Hughes JS, Bonazelli J, McCullough EC, Steinbeck BA, Mullin R, Tang AM, Muldoon J, Turner L et al (2003) All patient refined diagnosis related groups (APR-DRGs) version 20.0: methodology overview. Wallingford, CT: 3M Health Information Systems 91
16. Le Gall JR, Loirat P, Alperovitch A, Glaser P, Granthil C, Mathieu D, Mercier P, Thomas R, Villers D (1984) A simplified acute physiology score for ICU patients. Crit Care Med 12(11):975–977
17. Patel S, Hughes R, Hester T, Stein J, Akay M, Dy JG, Bonato P (2010) A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology. Proc IEEE 98(3):450–461. https://doi.org/10.1109/JPROC.2009.2038727
18. Hripcsak G, Albers DJ (2013) Next-generation phenotyping of electronic health records. J Am Med Inform Assoc: JAMIA 20(1):117–21. https://doi.org/10.1136/amiajnl-2012-001145
19. Bellifemine F, Fortino G, Giannantonio R, Gravina R, Guerrieri A, Sgroi M (2011) SPINE: A domain-specific framework for rapid prototyping of WBSN applications. Softw: Pract Exp 41(3):237–265. https://doi.org/10.1002/spe.998
20. Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml
21. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE (2000) Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation 101(23):215–220. https://doi.org/10.1161/01.CIR.101.23.e215

22. Kandhari R (2009) Anomaly detection. ACM Comput Surv 41(3):1–6. https://doi.org/10.1145/1541880.1541882
23. Banaee H, Ahmed MU, Loutfi A (2013) Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges. Sensors 13(12):17,472–17,500. https://doi.org/10.3390/s131217472
24. Ellis RJ, Zhu B, Koenig J, Thayer JF, Wang Y (2015) A careful look at ECG sampling frequency and R-peak interpolation on short-term measures of heart rate variability. Physiol Meas 36(9):1827–1852. https://doi.org/10.1088/0967-3334/36/9/1827
25. Li Q, Mark RG, Clifford GD (2008) Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a kalman filter. Physiol Meas 29(1):15–32. https://doi.org/10.1088/0967-3334/29/1/002
26. Pan J, Tompkins WJ (1985) A real-time QRS detection algorithm. IEEE Trans Biomed Eng 32(3):230–236. https://doi.org/10.1109/TBME.1985.325532
27. Rubin DB (1976) Inference and missing data. Biometrika 63(3):581–592. https://doi.org/10.2307/2335739
28. Wagstaff DA, Kranz S, Harel O (2009) A preliminary study of active compared with passive imputation of missing body mass index values among non-hispanic white youths. Am J Clin Nutr 89(4):1025–1030. https://doi.org/10.3945/ajcn.2008.26995
29. Camm AJ, Malik M, Bigger J, Breithardt G, Cerutti S, Cohen R, Coumel P, Fallen E, Kennedy H, Kleiger R et al (1996) Heart rate variability: standards of measurement, physiological interpretation, and clinical use. Eur Heart J 93(5):1043–1065. https://doi.org/10.1161/01.CIR.93.5.1043
30. Hampton JR (2013) The ECG made easy. Elsevier, New York
31. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238. https://doi.org/10.1109/TPAMI.2005.159
32. Moody GB, Mark RG (2001) The impact of the MIT-BIH arrhythmia database. IEEE Eng Med Biol Mag 20(3):45–50. https://doi.org/10.1109/51.932724
33. Eekhout I, de Boer RM, Twisk JW, de Vet HC, Heymans MW (2012) Missing data: a systematic review of how they are reported and handled. Epidemiology 23(5):729–732. https://doi.org/10.1097/EDE.0b013e3182576cdb
34. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Netw 12(2):181–201. https://doi.org/10.1109/72.914517
35. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
36. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27. https://doi.org/10.1145/1961189.1961199
37. Zweig MH, Campbell G (1993) Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clin Chem 39(4):561–577
38. Hillman K, Chen J, Cretikos M, Bellomo R, Brown D, Doig G, Finfer S, Flabouris A, Investigators MS et al (2005) Introduction of the medical emergency team (MET) system: a cluster-randomised controlled trial. Lancet 365(9477):2091–2097. https://doi.org/10.1016/S0140-6736(05)66733-5
39. Sun J, Wang F, Hu J, Edabollahi S (2012) Supervised patient similarity measure of heterogeneous patient records. ACM SIGKDD Explorations Newsletter 14(1):16–24. https://doi.org/10.1145/2408736.2408740
40. Hashem I, Yaqoob I, Anuar N, Mokhtar S, Gani A, Ullah Khan S (2015) The rise of big data on cloud computing: review and open research issues. Inf Syst 47:98–115. https://doi.org/10.1016/j.is.2014.07.006
41. Hossin M, MN S (2015) A review on evaluation metrics for data classification evaluations. Int J Data Mining Knowl Manag Process 5(2):1–11. https://doi.org/10.5121/ijdkp.2015.5201
42. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2012) Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: International workshop on ambient assisted living, pp 216–223. https://doi.org/10.1007/978-3-642-35395-6_30
43. R Core Team (2013) R: a language and environment for statistical computing r foundation for statistical computing, Vienna, Austria
44. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1):10–18. https://doi.org/10.1145/1656274.1656278b

45. Gattinoni L, Radrizzani D, Simini B, Bertolini G, Ferla L, Mistraletti G, Porta F, Miranda DR et al (2004) Volume of activity and occupancy rate in intensive care units. association with mortality. Intensive Care Med 30(2):290–297. https://doi.org/10.1007/s00134-003-2113-4
46. Iapichino G, Mistraletti G, Corbella D, Bassi G, Borotto E, Miranda DR, Morabito A (2006) Scoring system for the selection of high-risk patients in the intensive care unit. Crit Care Med 34(4):1039–1043. https://doi.org/10.1097/01.CCM.0000206286.19444.40
47. Barlow H (1989) Unsupervised learning. Neural Comput 1(3):295–311. https://doi.org/10.1162/neco.1989.1.3.295
48. Greene D, Cunningham P, Mayer R (2008) Unsupervised learning and clustering. Springer, Berlin. https://doi.org/10.1007/978-3-540-75171-7_3
49. Gravina R, Alinia P, Ghasemzadeh H, Fortino G (2016) Multi-sensor fusion in body sensor networks: state-of-the-art and research challenges. Inform Fusion 35:68–80. https://doi.org/10.1016/j.inffus.2016.09.005
50. Liggins MII, Hall D, Llinas J (2008) Handbook of multisensor data fusion: theory and practice. CRC Press, Boca Raton
51. Yang GZ, Hu X (2006) Multi-sensor fusion. Springer, London. https://doi.org/10.1007/1-84628-484-8_8
52. Chen C, Jafari R, Kehtarnavaz N (2015) A survey of depth and inertial sensor fusion for human action recognition. Multimed Tool Appl 76(3):1–21. https://doi.org/10.1007/s11042-015-3177-1
53. Graf AB, Smola AJ, Borer S (2003) Classification in a normalized feature space using support vector machines. IEEE Trans Neural Netw 14(3):597–605. https://doi.org/10.1109/TNN.2003.811708
54. Heart and Stroke (2013) The canadian heart and stroke foundation. Heart disease recovery road http://www.heartandstroke.com
55. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Tech. rep., DTIC Document
56. Depari A, Flammini A, Sisinni E, Vezzoli A (2014) A wearable smartphone-based system for electrocardiogram acquisition. In: IEEE international symposium on medical measurements and applications (MeMeA'14), pp 1–6. https://doi.org/10.1109/MeMeA.2014.6860030
57. Bellos CC, Papadopoulos A, Rosso R, Fotiadis DI (2010) Extraction and analysis of features acquired by wearable sensors network. In: 10th IEEE international conference on information technology and applications in biomedicine (ITAB'10), pp 1–4. https://doi.org/10.1109/itab.2010.5687761
58. Guidi G, Pettenati MC, Melillo P, Iadanza E (2014) A machine learning system to improve heart failure patient assistance. IEEE J Biomed Health Inform 18(6):1750–1756. https://doi.org/10.1109/JBHI.2014.2337752
59. Clifton L, Clifton DA, Pimentel MA, Watkinson PJ, Tarassenko L (2014) Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors. IEEE J Biomed Health Inform 18(3):722–730. https://doi.org/10.1109/jbhi.2013.2293059
60. Oresko JJ, Jin Z, Cheng J, Huang S, Sun Y, Duschl H, Cheng AC (2010) A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. IEEE Trans Inf Technol Biomed 14(3):734–740. https://doi.org/10.1109/titb.2010.2047865
61. Anliker U, Ward JA, Lukowicz P, Troster G, Dolveck F, Baer M, Keita F, Schenker EB, Catarsi F, Coluccini L et al (2004) AMON: a wearable multiparameter medical monitoring and alert system. IEEE Trans Inf Technol Biomed 8(4):415–427. https://doi.org/10.1109/titb.2004.837888
62. Kunnath AT, Nadarajan D, Mohan M, Ramesh MV (2013) Wicard: a context aware wearable wireless sensor for cardiac monitoring. In: International conference on advances in computing, communications and informatics, pp 1097–1102. https://doi.org/10.1109/ICACCI.2013.6637330
63. Solar H, Fernández E, Tartarisco G, Pioggiam G, Cvetković B, Kozina S, Luštrek M, Lampe J (2013) A non invasive, wearable sensor platform for multi-parametric remote monitoring in CHF patients. Health Technol 3(2):99–109. https://doi.org/10.1007/978-3-642-30779-9_18
64. Liu N, Lin Z, Koh Z, Huang GB, Ser W, Ong MEH (2011) Patient outcome prediction with heart rate variability and vital signs. J Signal Process Syst 64(2):265–278. https://doi.org/10.1007/s11265-010-0480-y
65. Leite C, Sizilio G, Neto A, Valentim R, Guerreiro A (2011) A fuzzy model for processing and monitoring vital signs in ICU patients. Biomed Eng Online 10:68–85. https://doi.org/10.1186/1475-925X-10-68

66. Bellos C, Papadopoulos A, Rosso R, Fotiadis DI (2011) A support vector machine approach for categorization of patients suffering from chronic diseases. In: Wireless mobile communication and healthcare, Springer, pp 264–267. https://doi.org/10.1007/978-3-642-29734-2_36
67. Gao H, Duan X, Guo X, Huang A, Jiao B (2013) Design and tests of a smartphones-based multi-lead ECG monitoring system. In: 35th international conference of the ieee engineering in medicine & biology society, pp 2267–2270. https://doi.org/10.1109/embc.2013.6609989
68. Kailanto H, Hyvarinen E, Hyttinen J (2008) Mobile ECG measurement and analysis system using mobile phone as the base station. In: Second international conference on pervasive computing technologies for healthcare, pp 12–14. https://doi.org/10.1109/PCTHEALTH.2008.4571014
69. Shih DH, Chiang HS, Lin B, Lin SB (2010) An embedded mobile ECG reasoning system for elderly patients. IEEE Trans Inf Technol Biomed 14(3):854–865. https://doi.org/10.1109/titb.2009.2021065
70. Pandian P, Mohanavelu K, Safeer K, Kotresh T, Shakunthala D, Gopal P, Padaki V (2008) Smart vest: wearable multi-parameter remote physiological monitoring system. Med Eng Phys 30(4):466–477. https://doi.org/10.1016/j.medengphy.2007.05.014
71. Juen J, Cheng Q, Schatz B (2015) A natural walking monitor for pulmonary patients using mobile phones. IEEE J Biomed Health Inform 19(4):1399–1405. https://doi.org/10.1109/jbhi.2015.2427511
72. Esfandiari N, Babavalian MR, Moghadam AME, Tabar VK (2014) Knowledge discovery in medicine: current issue and future trend. Expert Syst Appl 41(9):4434–4463. https://doi.org/10.1016/j.eswa.2014.01.011
73. Bellos CC, Papadopoulos A, Rosso R, Fotiadis DI (2014) Identification of COPD patients' health status using an intelligent system in the CHRONIOUS wearable platform. IEEE J Biomed Health Inform 18(3):731–738. https://doi.org/10.1109/jbhi.2013.2293172
74. Krause A, Ihmig M, Rankin E, Leong D, Gupta S, Siewiorek D, Smailagic A, Deisher M, Sengupta U (2005) Trading off prediction accuracy and power consumption for context-aware wearable computing. In: Ninth IEEE international symposium on wearable computers, pp 20–26. https://doi.org/10.1109/ISWC.2005.52