A Health Decision Support System for Disease Diagnosis Based on Wearable Medical Sensors and Machine Learning Ensembles

Hongxu Yin, *Student Member, IEEE* and Niraj K. Jha, *Fellow, IEEE*

Abstract—Even with an annual expenditure of more than $3 trillion, the U.S. healthcare system is far from optimal. For example, the third leading cause of death in the U.S. is preventable medical error, immediately after heart disease and cancer. Computer-based clinical decision support systems (CDSSs) have been proposed to address such deficiencies and have significantly improved clinical practice over the past decade. However, they remain limited to clinics and hospitals, and do not take advantage of patient data that are obtained on a daily basis using wearable medical sensors (WMSs) that have the ability to bridge this information gap. WMSs can collect physiological signals from anyone anywhere anytime. Thus, they have the potential to usher in an era of pervasive healthcare. However, most prior work on WMSs only focuses on hardware and protocol design, and not on an information system that can fully utilize the collected signals for efficient disease diagnosis. In this paper, for the first time, we introduce a hierarchical health decision support system for disease diagnosis that integrates health data from WMSs into CDSSs. The proposed system has a multi-tier structure, starting with a WMS tier, backed by robust machine learning, that enables diseases to be tracked individually by a disease diagnosis module. We demonstrate the feasibility of such a system through six disease diagnosis modules aimed at four ICD-10-CM disease categories. We show that the system is scalable using five more disease categories. Just the WMS tier offers impressive diagnostic accuracies for various diseases: arrhythmia (86 percent), type-2 diabetes (78 percent), urinary bladder disorder (99 percent), renal pelvis nephritis (94 percent), and hypothyroid (95 percent). We estimate that the disease diagnosis modules of all known 69,000 human diseases would require just 62 GB of storage space in the WMS tier. This is practical even in today's cloud or base station oriented WMS systems.

Index Terms—Clinical decision support, disease diagnosis, machine learning, pervasive healthcare, wearable medical sensors

1 Introduction

Fostered by modern healthcare, human life expectancy has increased by five years in the past two decades [1]. The healthcare system nurtures both physical and mental health of the population through clinical services and physician expertise, along with advancements in drug and prescription management. However, it also faces significant challenges. With more than 30 million annual admissions to registered hospitals, U.S. healthcare spending reached $3.2 trillion in 2015, and is predicted to reach $5.4 trillion by 2024 [2]. This has spurred enormous interdisciplinary research efforts from various research communities to improve the quality of healthcare.

Despite years of remarkable progress, the healthcare system is still far from being optimal. Some of it is due to challenges like lack of adequate cancer treatments that arise from our limited knowledge of the fundamental science involved. However, a substantial amount of sub-optimality in our current healthcare system can actually be avoided. Preventable medical errors (PMEs) fall under this category. A recent study claims that PME is the third leading cause of death in U.S. hospitals, with more than 250,000 annual deaths, immediately after heart disease and cancer [3]. Although whether these deaths are preventable has been questioned, the consensus remains that PMEs have a severe detrimental impact on patients. Moreover, since these studies only take inpatient records into account, the actual impact of PMEs may be much worse. Therefore, effective methods need to be deployed to reduce the occurrence of PMEs.

Reducing PMEs through human inspections consumes a huge amount of effort with only weak impact. The dominant cause of PMEs is poorly-designed clinical information systems [4]. More than 50 percent of such errors are found intimately linked to insufficient patient and drug information. Thus, computerized information systems, e.g., clinical decision support systems (CDSSs) and electronic health records (EHRs), have attracted significant research interest in the past decade. Long-term studies have shown that more than 66 percent of EHR-based CDSSs have significantly improved clinical practice [5]. As a result, healthcare organizations are now adopting these systems to assist physicians and healthcare providers with intelligently filtered clinical decisions. This adoption has been further sped up by the Health Information Technology for Economic and Clinic Health (HITECH) Act of 2009, which announced a $27 B federal government disbursement. Consequentially, an increasing amount of patient-specific clinical records is being collected every day, forming a fertile resource for data-oriented decision support systems.

• *The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544. E-mail: {hongxuy, jha}@princeton.edu.*

Manuscript received 17 Dec. 2016; revised 17 Apr. 2017; accepted 23 May 2017. Date of publication 31 May 2017; date of current version 13 Dec. 2017. (Corresponding author: Hongxu Yin.)
Recommended for acceptance by R. Marculescu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMSCS.2017.2710194

However, current CDSSs are still restricted to the clinical domain. CDSSs have very limited access to patient's health status when he/she leaves the clinic. This results in several deficiencies faced by today's CDSSs. Patients cannot always remember all the previous disease symptoms, which is an extremely important and sometimes the only information source for physicians and CDSSs for making decisions. This situation is made worse by the fact that many symptoms are not even noticeable. This points to the need for reliable, accurate, and intelligent out-of-clinic decision support complement for current CDSSs.

Another challenge is the non-uniformity of diagnosis offered by doctors. A typical training of a doctor in the U.S. takes more than seven years, with another three to seven years to acquire enough experience to make accurate decisions. Even though sharing of experience among doctors is possible through academic conferences and global summits, the standard deviation in doctor prescriptions is still quite large [6].

The emergence of WMSs points towards a way to address the above challenge. In the past 10 years, advancements in low-power sensors and signal processing techniques have led to many disruptive WMSs [7], [8], [9], [10], [11], [12], [13]. WMSs sense physiological signals passively and continuously. Useful health inferences can be derived from these signals. These sensors form a powerful, yet user-transparent, human-machine interface for tracking the health condition of the user. Hence, WMSs are viewed as a promising mechanism for enabling pervasive healthcare, thus forming a suitable complement to CDSSs in a daily context.

Unfortunately, a comprehensive WMS-based information framework for health monitoring of multiple diseases is still nonexistent, greatly impeding the impact of WMSs in CDSSs. Most current WMSs utilize a simple multi-threshold method to send alerts whenever signals fall outside the specified range. However, this approach is too weak to capture enough information for the more challenging task of medical diagnosis. Diagnosis requires a stronger information extraction mechanism, such as machine learning.

In this paper, we propose a hierarchical health decision support system (HDSS). Its novelty lies in the combination of WMSs and CDSSs through a hierarchical multi-tier structure supported by robust machine learning tiers. HDSS tackles both in- and out-of-clinic situations in a closed-loop manner. It hierarchically and sequentially structures the information framework for daily health monitoring, automatic symptom recording, accurate clinical diagnostic support, and post-diagnostic clinical support.

Our key contributions can be summarized as follows:

1) We present HDSS, a closed-loop multi-tier health decision support system that combines WMSs and CDSSs, in which the information framework is carefully structured in each tier.
2) We propose a recorder to store relevant raw data for symptoms to bridge the clinical information gap. With digitized memory, the problem of unreliable patient recall of symptoms can be addressed.
3) We present a scalable disease-module-based approach for monitoring disease. Each disease is individually tracked by its *disease diagnosis module* (DDM). Multiple DDMs generate disease *signatures* in parallel to track multiple diseases simultaneously. We also propose a procedure for automatically generating DDMs from biomedical datasets.
4) We hypothesize that different diseases (or at least disease classes) have different *signatures* that can be pervasively, accurately, and efficiently obtained from the physiological signals collected by WMSs. We provide support for this hypothesis through five accurate DDMs that utilize WMS data.
5) We evaluate scalability and storage requirements of HDSS. Our analysis shows that the DDMs for all reported 69,000 human diseases would only require around 62 GB of storage in the WMS tier. This is practical for both cloud and base station based WMS systems.

The rest of the paper is organized as follows. Section 2 provides background material required for understanding the rest of the paper. Section 3 presents design motivations through analysis of related work from the literature. Section 4 gives details of the proposed HDSS structure. Section 5 presents HDSS evaluation results. Section 6 discusses the limitations of this paper and future research opportunities. Finally, Section 7 concludes the paper.

2 Background

In this section, we discuss background material necessary for understanding the proposed HDSS structure. We introduce the concept of CDSS and ICD-10-CM code in Section 2.1. We present background material on WMSs in Section 2.2. Finally, we discuss the machine learning algorithms (MLAs) used in this paper in Section 2.3.

2.1 Clinical Decision Support Systems and ICD-10-CM

A CDSS is an active knowledge system that stores patient-specific data and generates case-specific suggestions. It is widely used in hospitals and clinics all over the world, supported by well-developed commercial platforms, such as TheraDoc, Safety Surveillor, QC PathFinder, Sentri7, and MedMined [5], [14]. The primary goal of a CDSS is to help clinician adherence to suggested medical guidelines, facilitate communications with hospitalized patients, enable secure access to patient medical data, and improve the quality of general healthcare service. A CDSS is usually built alongside or directly into a local EHR, possibly employing additional input resources, e.g., picture archiving and communication systems, computerized physician order entry, e-prescribing, and positive-identification medication administration systems. Upon generating useful insights, it sends message reminders, medical orders, prompts, alerts, suggestions or dashboards to physicians.

CDSSs and EHRs rely on coding systems to track diseases electronically. To construct a coding standard, World Health Organization introduced the International Statistical Classification of Diseases and Related Health Problems (ICD) coding system. Over the past several decades, ICD codes have successfully recorded diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injuries. Its latest version, the 10th revision (ICD-10), is a state-of-the-art coding system used by health entities around the world. The ICD-10 used in the U.S. has two major sections: ICD-10-CM for diagnosis coding and ICD-10-PCS for in-patient procedure coding. ICD-10-CM contains approximately 69,000 disease diagnosis codes covering 20 disease categories [15]. In this paper,

we use ICD-10-CM as the reference for disease diagnosis analysis in HDSS.

2.2 Wearable Medical Sensors

Due to rapid advancements in low-power sensing, computing, and communication, battery-powered WMSs are becoming increasingly ubiquitous. According to a report from BusinessInsiders, more than 56 million wearable sensors were sold worldwide in 2015, and this number is projected to increase to 123 million by 2018 [16]. These sensors capture, store, and transmit physiological data quietly, effectively, and efficiently. The list of collectable physiological signals include, but is not limited to, heart rate (HR), body temperature (BT), respiration rate (RESP), blood pressure (BP), electroencephalogram (ECG), electrocardiogram (EEG), Galvanic skin response (GSR), oxygen saturation (SpO_2), blood glucose (BG), and body mass index (BMI) [7], [8]. This list is expanding rapidly, given the speed of ongoing technological advancements in this field [17].

There is a rapidly growing body of literature on WMS based system design. A wrist-band based sensing platform, called AMON, can monitor key physiological data like BP, SpO_2, and ECG [9]. Similarly, LOBIN, an e-textile based WMS, has been validated in a clinical setup for sensing ECG, HR, and BT [10]. BlueBox collects ECG and bio-impedance through a novel hand-held device [11]. Emerging smart vests incorporate inter-fabric sensor pads that can actively sense multi-lead ECG, HR, BP, and GSR with moderate data quality and decent battery life [12], [13].

WMS based body-area networks (BANs) have also drawn enormous research attention. Communication protocols, transmission bandwidths, and secure encryption strategies have been discussed and analyzed for BANs [18]. In the CodeBlue project, the network is carefully designed to transmit vital health signs to health providers through secure routers [19]. The MobiHealth project explores an end-to-end mobile health monitoring platform with security, communication, and quality-of-service guarantees [20].

State-of-the-art BANs enable a wide range of healthcare applications. The feasibility of using WMSs for health monitoring has been verified in various healthcare domains, such as hypertension monitoring [21] and fitness tracking [22]. Using alert systems, these WMS based systems can make emergency calls, send messages, emails, and reminders to the care providers, depending on the severity of the emergency.

2.3 Machine Learning Algorithms

MLAs enable computers to learn through building of analytical models. When MLAs are guided by human labeling of data instances into various classes, they fall into the category of supervised learning. Supervised MLAs make predictions using mathematical rules learned from a labeled training dataset. Each training data instance contains a feature vector and its label as a target output. A feature represents a unique measurable phenomenon based on direct observation. Effective extraction of informative features is very challenging but extremely crucial for MLAs. For example, the multifractal dynamics in the ECG signal make it very hard to accurately model heart beats using the conventional R-R interval feature [23]. However, wavelet features can reduce such inter-patient variability and generate a more indicative model for pacemaker control through exploration of the spatiotemporal properties of ECG signals [24]. To avoid feature redundancy, a feature filtering technique can be used to search, evaluate, and select only the informative features.

TABLE 1
Machine Learning Algorithms and Ensemble Methods

Name	Abbr.	Descriptions
Naive Bayes	NB	Bayes theorem based probabilistic learner
Bayes network	BN	Network driven, conditional tables at nodes
k-nearest neighbor	IB-k	Similarity analysis with k closest instances
Best-first decision tree	BFTree	Tree with binary splits on features
J48	J48	Pruned or unpruned decision tree
Decision table	DT	A decision table based majority learner
Support vector machine	SVM	Support vector based linear separator
Multilayer perceptron	MLP	Back propagation based neural network
Stacker	ST	Combiner based stacking of base leaners
AdaBoost (Booster)	ADA	Weighted decision of weak classifiers
DECORATE (Voter)	DEC	Voting through diversified base learners
Bagger	BAG	Training with sampled subsets
Random tree	RT	Bagging on tree sampling features
Random forest	RF	Bagging on tree sampling instances/features

The primary goal of a supervised MLA is to mathematically map feature vectors into their corresponding labels, and store this rule for future predictions. When labels represent discrete classes (continuous values) of feature vectors, the problem is referred to as classification (regression).

In this paper, we primarily focus on classification problems with supervised MLAs due to their usefulness in the healthcare domain. We experiment with eight most widely used supervised MLAs and six ensemble methods. Table 1 shows their name and abbreviation, along with a short description. The upper section of Table 1 shows the MLAs used in this paper: Naive Bayes, Bayes network, k-nearest neighbor, best-first decision tree, J48, decision table, support vector machine (SVM), and multi-layer perceptron [25]. These representative MLAs cover three major machine learning categories: similarity based, probabilistic, and error based. Similarity based MLAs predict the label of an incoming data instance by analyzing its similarity to pre-known data instances: k-nearest neighbor predicts the class (value) of the unknown instance as the vote (mean) of most similar k instances, measured by similarity indicators such as Euclidian distance or inner product similarity in the feature space. Probabilistic MLAs predict the label of an incoming instance based on probabilistic relationships between feature values and labels: Naive Bayes utilizes the Bayes theorem to predict the class with the highest probability given a label-feature probabilistic relationship. Bayes network simplifies the lengthy Bayes chain calculation by introducing conditional tables. It increases the computational speed, at the expense of model size overhead. The best-first decision

tree is a binary tree that splits on feature values. Selection of the best feature type to split on depends on its ability to reduce impurities, typically measured by reduction in entropy levels, termed as information gain. J48 is another tree-based algorithm in which pruning algorithms are used to avoid overfitting. Decision tables utilize rule tables to represent the probabilistic relationship between features. Error-based MLAs generate mathematical models by reducing modeling errors based on various cost functions: SVMs construct hyperplanes to separate data in high-dimensional feature spaces. They use support vectors to maximize margins in order to reduce modeling error. A multilayer perceptron uses back propagations to train a neural network structure that minimizes an objective cost function, and predicts the label of new data instances through forward propagation.

As its name suggests, an ensemble method describes rules for a meta learner that makes a final decision based on an ensemble of decisions from base learners, e.g., the MLAs mentioned above. As a result, an ensemble method can significantly boost MLA performance [25]. In this paper, we explore the six ensemble methods shown in the lower section of Table 1. A stacker stacks various types of classifiers hierarchically where high-level classifiers correct the instances mislearnt by low-level classifiers. AdaBoost is the most widely used boosting algorithm, which uses a sequence of weak classifiers to generate a strong classifier. This is based on increasing the weights of previously misclassified data instances. DECORATE introduces diversification in the voters using artificial training examples sampled from the original dataset, but with labels inversely proportional to existing voter predictions. A new voter is added to the final voter pool only if it can pass the diversification test, where its classification results on this artificial training set differ from those of previous voters [26]. Bagger, or a bootstrap aggregating method, trains various base learners using subsets sampled from the original training set, and makes predictions based on various rules, such as voting and averaging. In the healthcare domain, the two most widely used bagging methods are random tree and random forest because of their accuracy and interpretability. Random tree samples the feature space, and random forest samples both the training instance space and feature space. Using a bootstrap sampling rule and feature splitting method, random tree and random forest can generate ensemble structures without an external base learner. Thus, they are different from other previously mentioned ensemble methods.

Depending on base learner types, ensemble methods can be divided into two categories: homogeneous and heterogeneous. A homogeneous method employs base learners of the same MLA type, whereas a heterogeneous method uses more than one MLA type.

3 Related Work and Motivation

As mentioned earlier, CDSSs are currently only employed in a clinic or hospital [5]. Outside this domain, the system has to rely on the ability of patients to recall their symptoms, which is quite error-prone. Second, not all disease symptoms are trackable by humans, thus cannot be effectively fed to CDSSs. For example, symptoms of arrhythmia or diabetes are not easily noticeable by an individual. They often become evident only when something severe occurs, e.g, abrupt poundings of the chest or consistent loss of weight. Thus, it is difficult to respond in a timely and effective manner, resulting in irreversible detrimental effects. These shortcomings of existing CDSSs suggest complementing them with WMS based disease diagnosis.

Unfortunately, WMS based disease diagnosis is yet to be comprehensively studied. Previously proposed alert systems employ a simple multi-threshold mechanism, in which thresholds are used to track abnormal signals [9], [21], [22], [27]. AMON issues alerts when the collected bio-signals fall outside a pre-defined range [9]. The mobile care alert system for hypertension risk analysis bins the monitored blood pressure values into four categories separated by thresholds [21]. Similarly, a web-based remote system assigns discrete risk levels for BMI, BP, and BG [27]. A real-time physiological analysis system for fitness tracking uses the same approach, but with a threshold-tuning method [22]. However, the multi-threshold method suffers from multiple shortcomings. First, the appropriate thresholds depend on the individual, location, and time [22]. Thus, training a robust, yet accurate, threshold model requires a lot of individualized effort. Second, such simple methods fail to capture enough information to enable more challenging disease diagnosis, as demonstrated later.

In this paper, we try to answer the basic question of whether WMSs can be used for disease diagnosis. We hypothesize that such a scheme is feasible, and we provide support for this hypothesis using a machine learning approach. Then, we study the feasibility of merging this powerful tool with existing CDSSs to show the possibility of forming a comprehensive and closed-loop pervasive healthcare support system: HDSS.

4 Closed-Loop Hierarchical HDSS Structure

In this section, we describe the proposed HDSS structure in detail. First, we discuss its hierarchical multi-tier structure. Then, we zoom into individual tiers and analyze their information frameworks. Finally, we show the detailed structure of the DDM used by HDSS to track each disease separately, followed by a discussion of the automatic DDM generation procedure.

The schematic diagram of HDSS is shown in Fig. 1. HDSS has two major parts, separated by the clinical boundary. The first part, shown on the left, is the pervasive health decision support (PHDS). The other part is PHDS-assisted CDSS, denoted as CDSS+. PHDS acts on data collected by WMSs for daily health monitoring, while CDSS+ assists with clinical decisions.

HDSS has four major tiers, as shown in Fig. 1. It uses these tiers to sequentially model general healthcare from daily health monitoring, initial clinical checkup, detailed clinical examination, and post-diagnostic treatment. Tier-1 assists with daily health monitoring. It incorporates decision modules trained using clinical domain knowledge, and transmits information across the clinical boundary. This helps individuals, even those without professional medical training, track their diseases. Tier-2 provides immediate decision support to physicians for an incoming patient. At this tier, physician insight and basic measurements provide additional inputs, even though very accurate laboratory results are not yet available. At Tier-3, a more detailed diagnostic analysis becomes feasible based on laboratory measurements. Finally, Tier-4 delivers post-diagnostic treatment, prescription, and lifestyle suggestions. HDSS operates from Tier-1 to Tier-4 in a sequential and closed-loop manner, as indicated by the large arrow connecting all four tiers

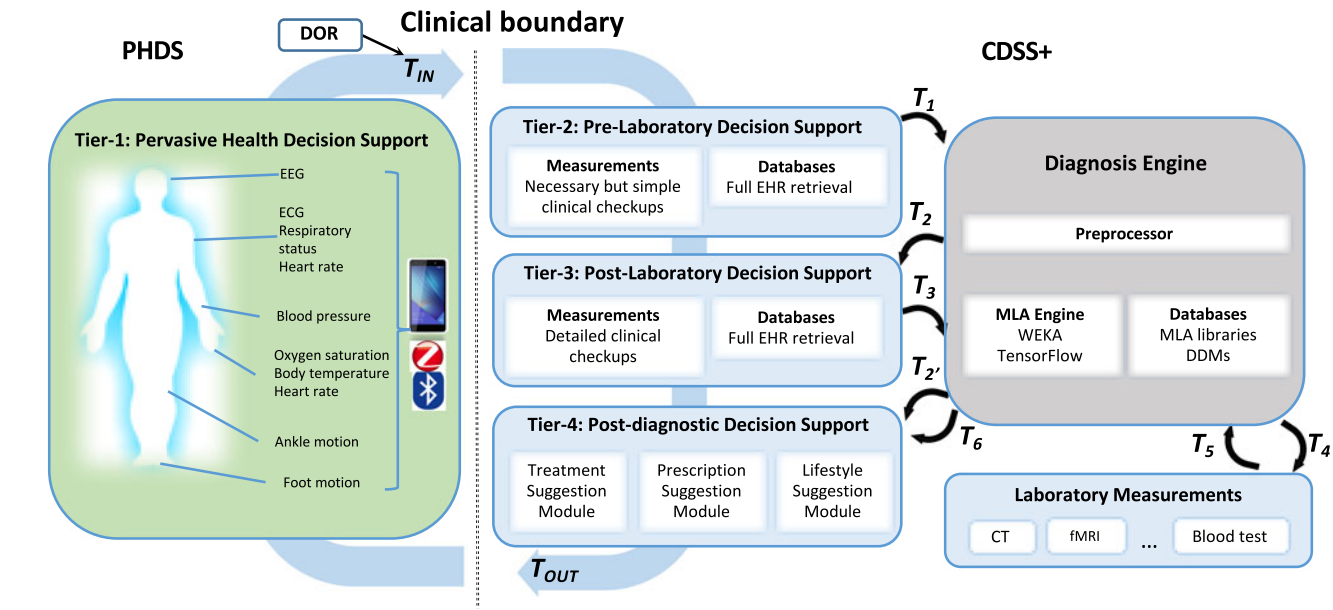


Fig. 1. Schematic diagram of HDSS with two components: Pervasive health decision support (PDHS) and PHDS-assisted clinical decision support system (CDSS+). Transition i, disease diagnosis modules, and disease-onset records are denoted by T_i, DDM, and DOR, respectively.

in Fig. 1 (note that the term *closed-loop* in this paper has a different meaning from the one used in feedback system design). Hence, operation and information flows in the loop are directional. Tier-x data are available to Tier-y, $x < y$, however, the reverse is not true. Thus, subsequent tiers gather more information than previous tiers, but typically with higher time and energy costs.

To make sure HDSS can smoothly transfer information across tiers, the tiers are interconnected via transitions (T), depicted by indexed arrows in Fig. 1. When an alert is raised at Tier-1, a transition T_{IN} crosses the clinical boundary and transfers patient information to Tier-2. Along with T_{IN}, Tier-1 passes relevant symptom records stored as disease-onset records (DORs) for subsequent analysis. At Tier-2, the data, aggregated with additional measurements and physician insights, are passed to the diagnosis engine through T_1. In HDSS, we allocate all the decision-making processes to an external diagnosis engine, so that modifications needed to existing CDSSs can be minimized. The diagnosis engine contains MLA libraries and DDMs in databases accessible by MLA engines, such as WEKA and TensorFlow [28], [29]. Details of the disease-specific DDM structure are given later. There are two possible outgoing transitions from the diagnosis engine: T_2 and $T_{2'}$. When further laboratory measurements are needed to make a final diagnosis, T_2 transfers HDSS to Tier-3. Otherwise, HDSS reaches Tier-4 through $T_{2'}$. Regardless of which transition is selected, diagnostic suggestions from the diagnosis engine and information on whether further laboratory measurements are needed are immediately available to physicians at Tier-2. When T_2 occurs, Tier-3 calls the diagnosis engine through T_3. Appropriate laboratory tests are ordered through T_4, and reports fed back to the diagnosis engine through T_5. At this stage, a more detailed disease-specific diagnosis can be performed. Diagnostic analysis at Tier-3 is more time-consuming and expensive than at Tier-2. Laboratory measurements can be slow and expensive. For example, a blood test report can take as much as 12-16 hours, and reports for computed tomography (CT) and functional magnetic resonance imaging (fMRI) tests are even slower. However, Tier-3 is still the most important tier since it has all available information for making a concrete diagnosis. Tier-4 is reached through $T_{2'}$ or T_6. All data measurements, extracted features, and diagnostic suggestions from previous tiers are passed on to Tier-4 for making post-diagnostic suggestions. Upon a satisfactory outcome, a final transition, T_{OUT}, indicates completion of the clinical visit and transfers the HDSS state back to Tier-1 PHDS.

Next, we zoom into the tier-wise information framework.

4.1 Tier-1 Pervasive Health Decision Support

This tier uses WMS data to detect/track multiple diseases. As shown in Fig. 2, we model the diagnostic decision flow of PHDS using six sequential stages: selection of target physiological signals, matching of these signals with their WMSs, pre-processing of the collected signals for machine learning models (MLMs) pre-trained using MLAs, decision making through MLMs, obtaining disease signatures, and responding according to the decisions. Diagnosis of disease i is done through its own tier-wise disease module i, shown as a vertical dotted box in Fig. 2. Using this structure, PHDS can monitor n diseases in parallel.

In the first two stages, the p physiological signal types are matched with their corresponding WMSs through a signal-sensor look-up table (SS-LUT), as shown in Table 2. The signals are divided into B-series (physiological), M-series (motion), and L-series (location). A context recognizer, shown at the top of each disease module, uses the M-series and L-series data to validate the context for B-series data. For example, muscle movements can generate noise in ECG measurements. Consequentially, a clinical 12-lead ECG measurement requires the patient to stay still for at least 30 seconds. In Tier-1, however, such a constraint on the users is impractical. WMS data collected during fitness training may be too noisy for arrhythmia analysis, and thus may need to be flagged as such.

In the third stage, the digitized signals from WMSs are pre-processed from possibly incomplete and inconsistent raw measurements into formats understandable by MLMs. Upon approval from the context recognizer, the appropriate

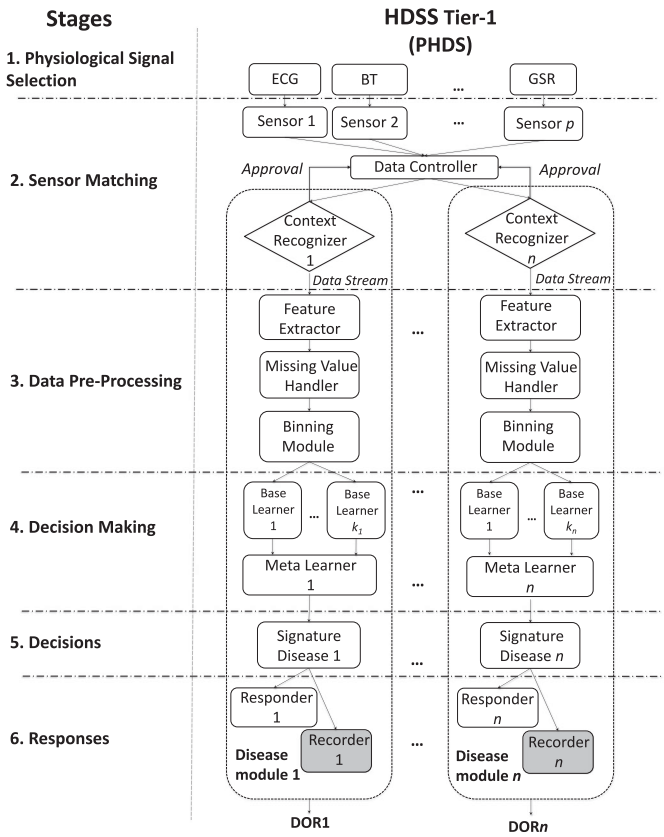


Fig. 2. Information framework of Tier-1 PHDS. Disease onset record of disease i is denoted as DORi.

data streams are fed to a disease module based on the SS-LUT indices specified by the module. The feature extractor computes the desired features from these data streams. Then, the missing value handler gets rid of missing feature values using statistical methods, such as value imputations and interpolations. Finally, the binning module maps continuous measurements to separate bins, thus reducing overall computational load on the MLMs.

Diagnostic decisions are made in the fourth stage. In PHDS, diagnosis of disease i is done using its pre-trained MLM that consists of a meta learner and k_i base learners. A meta learner finalizes a prediction based on k_i base learner outputs, using the ensemble method selected during the training phase. Note that sometimes the training phase may prefer a single base learner, thus obviating the need for a meta learner, given the fact that ensemble methods do not always boost MLM performance. In this case, we set $k_i = 1$ to store only the desired base learner, and the meta learner passes this information to the next stage.

The fifth stage obtains the binary signatures of the n diseases being targeted. A $signature$ of 0 indicates benign status, and a $signature$ of 1 indicates potential disease onset. For each disease module, a series of 1s triggers the responder and recorder of this disease in the sixth stage. The recorder stores the raw measurements that led to this disease diagnosis in a DOR for future clinical usage. The responder assists the users with appropriate medical suggestions for this disease, and directs them to the most relevant part of the clinic. This transition from PHDS to clinic saves an enormous amount of time for both physicians and patients by completely bypassing the lengthy initial status-checking procedure performed in the clinic.

Table 3 presents the cost of sensing and storing seven common physiological signals used by the PHDS. We have obtained these values from WMS based long-term continuous health monitoring scheme presented in [30]. We can see that both the energy and storage costs are modest and well within the capabilities of current technologies. Note that $signature$-driven DORs only need to store raw measurements that trigger disease diagnosis. Since this can be expected to be quite infrequent, these values in fact only indicate a cost upper bound.

4.2 Tier-2 Pre-Laboratory Clinical Decision Support

This tier helps physicians evaluate incoming patients in a clinical setup. Fig. 3 shows the parallel decision flows of Tier-2 and the physician. The decision flows have five stages.

TABLE 2
Signal-Sensor Look-Up Table (SS-LUT) for Tier-1 PHDS

Index	Signals	Abbreviations	Data Type	Unit	Matched Sensor
B1	Arterial systolic blood pressure	ABPSYS	float	$mmHg$	Cuff-less BP monitor (wrist/arm)
B2	Arterial diastolic blood pressure	ABPDIAS	float	$mmHg$	Cuff-less BP monitor (wrist/arm)
B3	Blood glucose	BG	float	mg/dL	Laser-based wearable sensor, wrist sweat sensor
B4	Body temperature	BT	float	$Celsius$	Watch
B5	Electroencephalogram	EEG	string	μV	Hat
B6	Electrocardiogram	ECG	string	μV	Vest, chest-band
B7	Galvanic skin response	GSR	string	μS	Ring+wrist-band, gloves
B8	Heart rate	HR	float	$1/min$	Wrist-band, chest-band, smart watch, ring
B9	Oxygen saturation	SpO_2	float	%	Ring
B10	Pulmonary systolic artery pressure	PAPSYS	float	$mmHg$	Vest
B11	Pulmonary diastolic artery pressure	PAPDIAS	float	$mmHg$	Vest
B12	Respiratory rate	RESP	float	$1/min$	Chest-band
M1	Ankle acceleration	accAnkle	string	m/s^2	Ankle-band
M2	Body acceleration	accBody	string	m/s^2	Smartphone
M3	Hand acceleration	accHand	string	m/s^2	Wrist-band, smart watch
L1	Altitude	ALT	float	m	Smartphone
L2	GPS position	GPS	float	$degrees$	Smartphone
L3	Moisture	MOS	float	%	Smartphone

TABLE 3
Cost of Sensing and Storing Signals for PHDS

Bio-signal	Resolution (bits)	Sampling frequency (Hz)	Daily sensing energy upper bound (J)	Annual storage requirement upper bound (MB)
Heart Rate	10	2-8	14-55	75-297
Blood pressure	16	0.001-100	0.26-687	0.07-6,011
Oxygen saturation	8	0.001-2	0.26-14	0.03-61
Temperature	8	0.001-1	0.26-7	0.03-31
Blood sugar	16	0.001-100	0.26-687	0.07-6,011
ECG	12	100-1,000	687-6,869	4,511-45,117
EEG	12	100-1,000	687-6,869	4,511-45,117

During a patient's first clinical checkup, physicians extract background information on the patient through a series of questions, observations, and a review of EHR. The effectiveness of this information extraction process depends heavily on physician experience. Physicians may also utilize n Tier-2 tests and measurements to acquire more relevant disease indicators. The physician processes the aggregated information in his/her brain, and decides whether further Tier-3 tests are needed or if a diagnosis can be finalized. Due to different physician-specific styles of practice, this decision making process causes inter-physician practice variations [6].

Tier-2 has a similar decision flow. Disease diagnosis is carried out using the disease module shown in the vertical dotted box in Fig. 3. As opposed to the conventional information extraction process used by physicians, Tier-2 utilizes EHRs and DORs to inform the physicians of patient's background information and a detailed symptom record. The DOR symptom types can be updated based on latest domain knowledge to provide a uniform picture to all physicians. Tier-2 recommends m tests based on the collected empirical data. Since a physician may not remember all available m tests at a given time, we have $m \geq n$ for Tier-2 HDSS. As in the case of Tier-1, the test measurements are pre-processed before being fed to a meta learner through k base learners. Finally, Tier-2 transfers control to either Tier-3 or Tier-4 depending on whether further laboratory measurements are necessary.

4.3 Tier-3 Post-Laboratory Clinical Decision Support

In this tier, the main objective is to make an accurate diagnosis with help from all available medical measurements. Compared to previous tiers, diagnosis in this tier can be narrowed down to within a few similar disease sub-types. In other words, where Tier-1 and Tier-2 perform horizontal monitoring of many diseases, Tier-3 performs a deep vertical analysis into a specific disease candidate.

Tier-3 uses the same decision flow as Tier-2, however, with the information extraction stage removed and a modified clinical testing stage. The tests are now expanded from the immediate and simple clinical tests to all relevant complex, sophisticated, but very informative, laboratory tests. These tests may require extra assistance, processing time, equipment, and experiments, such as blood tests, CT, and fMRI. Though quite beneficial, these tests incur significantly higher time and expense. As a result, simplifying Tier-3 tests has attracted substantial research attention. For example, the SIMBAS technology from UC Berkeley is capable of reducing blood test time to ten minutes, thus pushing it from Tier-3 towards Tier-2 [31]. Therefore, the exact allocation of tests to these tiers needs to be updated periodically and, in fact, be decided based on facilities available in the clinic.

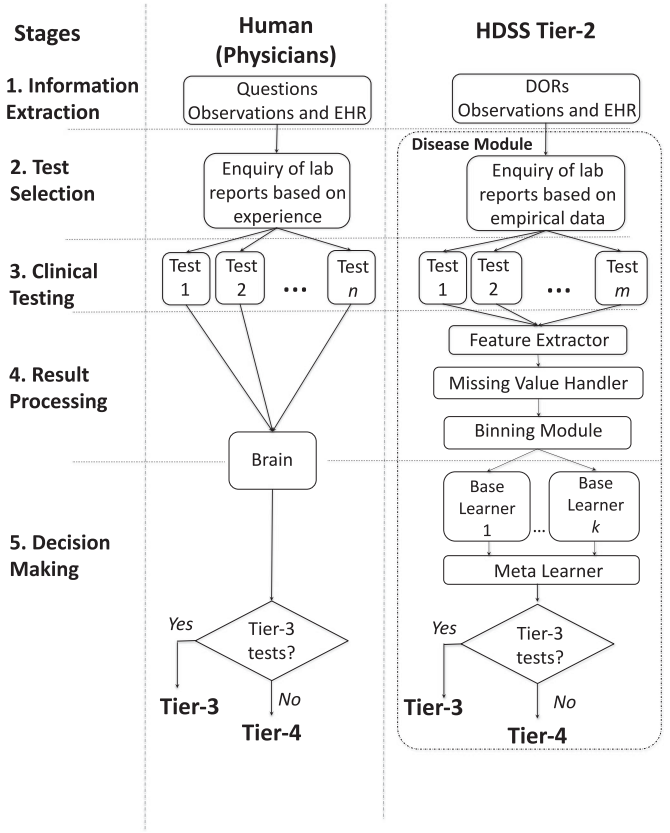


Fig. 3. Tier-2 information framework for clinical disease diagnosis with a parallel decision flow of a human physician.

4.4 Tier-4 Post-Diagnostic Decision Support

Tier-4 provides post-diagnostic decision support, where MLA-assisted treatments [32], prescriptions and medications [33], and future lifestyle suggestions [34] can be generated by the respective modules. In this tier, it is much harder to derive a unified decision flow or information framework, as these modules have different final objectives and serve various end-users. However, all these modules share the same need for diagnostic data from previous tiers.

Based on information from Tier-2 or Tier-3, if the health status of the patient is satisfactorily resolved, T_{OUT} transfers control across the clinical boundary. CDSS+ finalizes the clinical visit by appropriately updating EHRs, generating patient-centered lifestyle recommendations, such as saturated fat and cholesterol intake restrictions [34], and transferring HDSS back to the PHDS tier to initialize a new monitoring round.

4.5 Disease Diagnosis Modules

As mentioned earlier, diseases can be tracked by their independent DDMs in each tier. A DDM specifies the unique and necessary information framework components used by HDSS for diagnosis. Hence, to evaluate or update the diagnostic rule for a given disease, one only needs to modify its DDM instead of restructuring the entire HDSS. DDMs share the standardized information framework shown in Fig. 4. A DDM is composed of a unique code and its tier-wise disease modules. In order to be consistent with existing CDSSs and EHRs, the DDM code is set to the ICD-10-CM code of the disease. The disease module for disease i at Tier-j is denoted as Di-T-j, $j = 1, 2, 3$. Components shown

DDM Structure of Disease *i*

Code	ICD-10-CM of Disease *i*	
D*i*-T-1	Sensor Set Code T-1	
	Context Recognizer T-1	
	Preprocessor Set T-1	Feature extractor Set T-1 Binning module T-1 Missing value handler T-1
	Decision Maker T-1	Base learners Meta learner
	Responder and Recorder	
D*i*-T-2	Test Set Code T-2	↔ Indices of T-2 tests
	Preprocessor Set T-2	
	Decision Maker T-2	
D*i*-T-3	Test Set Code T-3	↔ Indices of T-3 tests
	Preprocessor Set T-3	
	Decision Maker T-3	

Fig. 4. The proposed DDM structure for disease i supported by the HDSS framework. Disease module for disease i at Tier-j is denoted by Di-T-j.

in green communicate with external electronic systems. The sensor set code works with SS-LUT to specify the WMSs needed for diagnosis of disease i. The test set code searches through the clinical test directory for desired laboratory measurements. The components in white correspond to responders and recorders. The components in grey represent the intelligent MLM sectors. A decision maker, which stores the MLM, makes diagnostic predictions based on the latest domain knowledge extractable from an up-to-date training dataset on disease i, and thus acts as the core of a DDM. The preprocessor set for a decision maker stores its corresponding feature extractor, binning module, and missing value handler. Once a decision maker is generated through the training process, its corresponding preprocessor set can be obtained simultaneously.

We use a DDM generation procedure to automatically generate DDMs. As shown in Fig. 5, we sequentially construct a training table from a biomedical dataset for disease i, derive tier-wise available datasets, generate decision makers in a parallel fashion, and finalize the DDM for disease i.

A training table for disease i can be acquired from a biomedical dataset after the feature indexing process is complete. Each feature is given an availability index in the form of an integer ranging from 1 to 3 to signal its availability at Tier-1 through Tier-3. A decision maker (DMx) at Tier-x can only be trained using features with indices t, where $t \leq x$. As shown by the vertical parentheses in Fig. 5, data instances containing available features at a given tier form an available dataset for that tier. This available dataset can later be used to generate the decision maker. Since the training table can simultaneously support multiple available datasets, decision makers can be generated in a completely parallel manner.

Given an available dataset, there are three major stages in decision maker generation, as shown in the middle section of Fig. 5. Stage 1 generates a set of base learners and their performance parameters. We utilize a performance matrix to store five important performance parameters for a base learner, as shown in Table 4. Accuracy (ACC) indicates the base learner's overall prediction capability. The true-positive rate (TPR) and true-negative rate (TNR) measure the base learner's capability to recognize disease and benign cases, respectively. The false positive rate (FPR) specifies the percentage of false alarms generated by the base learner. The F1 score measures the overall performance of TPR and TNR. The area under the curve (AUC) is another accuracy metric, which uses a receiver operating characteristic (ROC) curve to capture the tradeoffs between the TPR and FPR. A value close to 1 is preferred for ACC, TPR, TNR, F1, and AUC. Stage 2 generates a series of meta learners using base learners passed from Stage 1. To provide guidance to this process, performance matrices are checked by a checker to match base learner candidates with appropriate ensemble methods, e.g., AdaBoost prefers weak base learners and voters prefer diversified base learners. In Stage 3, a statistical selector compares performance matrices of all generated learners from both stages based on pre-defined statistical criteria, such as TPR, TNR, F1 score, McNemar metric, geometric mean error, and win/draw/loss game [35], and selects the best learner as the final decision maker.

The generated decision makers are packaged into the final DDM, together with their performance matrices, and their learning statistics. Storing the learning statistics is important to effectively defend against an adversarial machine learning attack, which aims to negatively impact the decision making process by degrading the performance of MLMs [36]. Finally, the complete DDM is stamped with its ICD-10-CM code, and uploaded to HDSS.

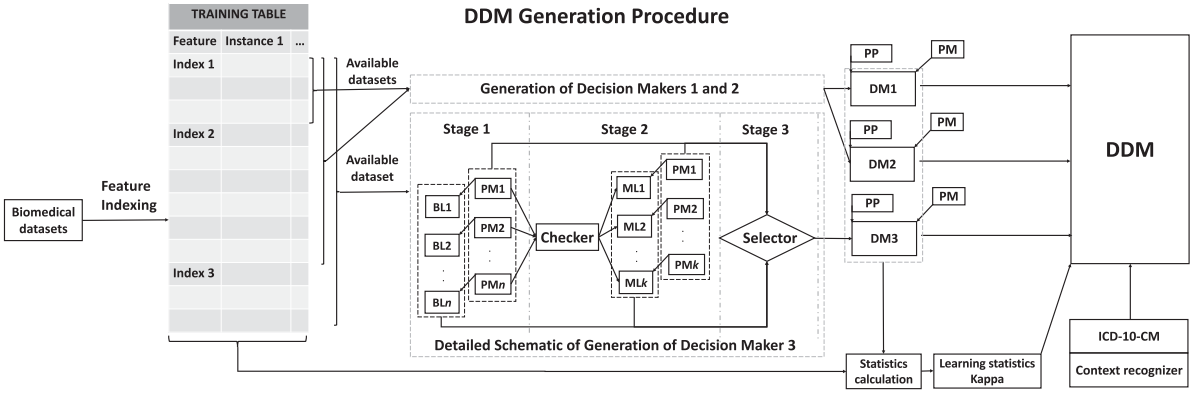


Fig. 5. The schematic flowchart of an automatic DDM training procedure that derives DMs from a biomedical dataset. The base learner, meta learner, performance matrix, decision maker, and pre-processor are denoted by BL, ML, PM, DM, and PP, respectively. Kappa statistic measures relative improvement over random predictors.

TABLE 4
Performance Matrix

Name	Definition
Accuracy (ACC)	$(TP+TN)/(TP+FP+FN+TN)$
True positive rate (TPR)	$TP/(TP+FN)$
True negative rate (TNR)	$TN/(TN+FP)$
False positive rate (FPR)	$FP/(TN+FP)$
F1 score (F1)	$2TP/(2TP+FP+FN)$
Area under the curve (AUC)	$\int ROC$

–TP (TN): disease (benign) instances classified as disease (benign)
–FP (FN): benign (disease) instances classified as disease (benign).

5 Evaluating HDSS Performance

In this section, we evaluate the performance of the proposed HDSS. We first show how an arrhythmia DDM can be constructed based on its biomedical dataset. We then evaluate the DDMs for five other diseases, thus demonstrating the scalability of HDSS and the feasibility of using WMSs for disease diagnosis. We also estimate the HDSS storage requirements for all 69,000 human diseases.

5.1 Arrhythmia DDM Generation

In this section, we use arrhythmia to show how the DDM generation procedure works. The arrhythmia dataset was acquired from the openly accessible UCI repository [37]. It contains 452 data instances, each of which stores 279 feature values extracted from a 12-lead ECG recording. Instances are either labeled as benign or one of 15 arrhythmia subtypes.

5.1.1 Training Table Acquisition

As mentioned earlier, we transform a given biomedical dataset into a training table with the feature indexing process. Feature indexing at Tier-1 depends on the available WMS types in SS-LUT that can be matched to the signals in the biomedical dataset. In this paper, we select the 12-lead smart vest as the WMS for ECG collection [13]. We pick this WMS to obtain the highest achievable accuracy in the WMS tier. With access to all 12 ECG channels in Tier-1, Tier-2, and Tier-3, the 279 features are available in all tiers. Hence, we label all the features with index 1. Note that for other ECG WMSs in the SS-LUT that use fewer leads, the feature indices may be different. For example, feature indexing for a three-lead ECG WMS group results in 81 features (4 demographic features, 11 overall interval features, and 22 features for each of the 3 channels in WMS tier) with index 1 and 198 features (22 features for each of the remaining 9 channels available in Tier-3) with index 3.

5.1.2 Decision Maker Generation

We implement the decision maker generation procedure using WEKA 3-7-13 [28]. Due to the binary classification performed in Tier-1, the labels in the available dataset for DM1 are re-mapped to binary indicators of arrhythmia existence or non-existence. The available datasets for Tier-2 and Tier-3 maintain their 16-class labels. Next, we explain the DM1 generation procedure in detail. DM2 and DM3 are generated in parallel using the same methodology.

In Stage 1, we generate eight base learners from the original available dataset of learners and another eight base learners from the feature-filtered available dataset. These 16 base learners form a base learner candidate pool for the selection of the final decision maker. The base learners include Naive Bayes, Bayes network, SVM, k-nearest neighbor, best-first decision tree, J48, decision table, and multilayer perceptron. Feature filtering is based on supervised forward feature selection [38]. Since feature filtering does not guarantee performance improvement, we pass these 16 base learners and their performance matrices to Stage 2 for meta learner generation and Stage 3 for DM1 generation. Unless otherwise stated, we use six-fold cross-validation to generate performance matrices. In a six-fold cross-validation, the original dataset is randomly partitioned into six subsets. In each evaluation round, a new MLM is trained using five subsets and evaluated on the remaining subset. Thus, six models are generated in six rounds. The average performance across the six generated models represents the final performance of this type of MLM.

In Stage 2, we use six ensemble methods to generate the meta learners. These methods include AdaBoost, bagger, voter, stacker, random forest, and random tree. We apply feature bagging to all 16 base learners. We use two types of voters: rule based and diversification based. The rule based voter combines base learners with six voting rules: average of probabilities, product of probabilities, majority voting, maximum probabilities, minimum probabilities, and median probabilities. This voter covers max, min, median, and majority rules for posterior probability calculations that generate a final prediction based on base learner classification results. We generate two separate learner pools for this type of voter, with the first learner pool containing eight base learners from the original set and the other containing eight base learners from the feature-filtered set. The second type of voter, DECORATE, introduces diversification in the learner pool by adding a new voter that disagrees with previous voters on an artificial sample pool. This sample pool contains training instances that are sampled from the original dataset with new labels generated in an inversely proportional fashion to existing predictions [26]. In our current setting, the diversity depth (the number of diversified voters generated) is set to 15, since we found this to be the point of diminishing returns. For stacker, we generate two stacking models using the same learner pools as rule-based voters. Random forest and random tree are implemented separately on these learner pools as well. They are considered homogeneous because they are independent of external base learners. All the generated meta learners from Stage 2 form a meta learner candidate pool for final decision maker generation.

In Stage 3, the final DM1 is obtained using a statistical selector that operates on the base learner and meta learner candidate pools based on a pre-defined selection criterion. We use classification accuracy for this purpose due to its general effectiveness and widespread use. However, another statistical criterion mentioned in Section 4.5 could also be used.

5.1.3 Performance Analysis

Fig. 6 shows the classification accuracy for the different methods. 'Single' represents a base learner that does not use feature filtering or an ensemble method, +F (+E) represents feature filtering (ensemble method), and +E+F represents both. We can see that accuracy can be enhanced through both feature filtering and ensemble methods in most cases. Sometimes, the combination of the two methods gives

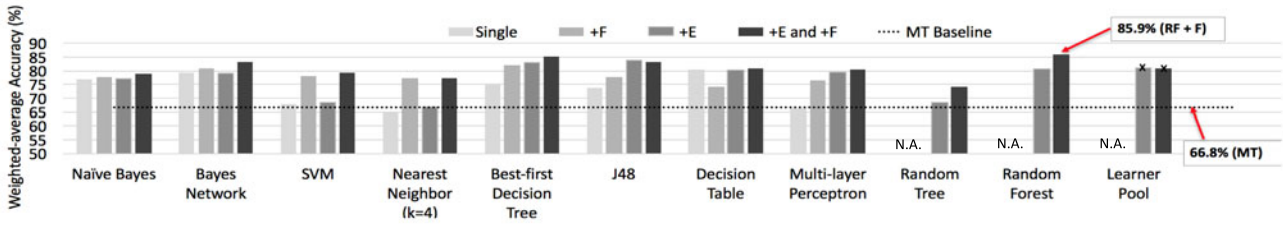


Fig. 6. Accuracy of the arrhythmia decision maker in the WMS tier using a single classifier, feature filtering (+F), ensemble method (+E), and both (+E +F). The 10 bar groups to the left use homogeneous ensemble methods based on AdaBoost, DECORATE, and bagger. The rightmost bar group (marked with ×) uses a heterogeneous ensemble method based on stacker and voter. N.A.: Not applicable.

the best accuracy. From this pool, the best accuracy (85.9 percent) is obtained for random forest with feature filtering (RF+F). Hence, it is selected as DM1.

To compare our machine learning approach against the conventional multi-threshold approach [9], [21], [22], [27] in the WMS-tier, we train a best-first decision tree model on each of the 279 Tier-1 features individually. This approach splits each possible feature axis into value ranges separated by thresholds and checks the rule's accuracy. In other words, it tries to find the highest accuracy the multi-threshold approach can realize by exhaustively constructing threshold models. The highest accuracy for the multi-threshold approach was found to be only 66.8 percent (denoted by the dashed MT baseline in Fig. 6). Thus, the multi-threshold approach is not very accurate.

5.1.4 Final Arrhythmia DDM

We show the arrhythmia DDM in Fig. 7. We use the same DDM generation procedure to obtain DM2 (ACC: 77.4 percent) and DM3 (ACC: 77.4 percent). Note that the drop in their accuracies compared to DM1 arises from their tackling of a more challenging 16-class classification task. We use the Kubios HRV Software to extract all 279 features from ECG signals [39]. In the WMS tier, the sensor set code is B6, which matches the 12-lead ECG vest in SS-LUT (see Table 2). The test set codes in Tier-2 and Tier-3 are set to null entry and 12-lead ECG test code, respectively, since clinical 12-lead ECG measurement is a Tier-3 test that requires technician assistance. To get rid of muscle noise, the context recognizer is set to 'still', which is within the monitoring capabilities of current WMSs [40]. Finally, the DDM is stamped with its ICD-10-CM code (I49.9), and uploaded to the HDSS framework.

Arrhythmia DDM

Code	I49.9
D-T-1	B6: 12-lead vest
	Context recognizer set as 'still'
	Kubios HRV [39]
	RF + F, ACC: 85.9% Bi-classification
	Clinical appointment with cardiovascular physicians
D-T-2	N.E.
	Kubios HRV [39]
	BAG(BN) + F, ACC: 77.4% Benign + 15 sub-types
D-T-3	12-lead ECG (clinical test)
	Kubios HRV [39]
	BAG(BN) + F, ACC: 77.4% Benign + 15 sub-types

Fig. 7. The final arrhythmia DDM. N.E.: Null entry. RF: Random forest. +F: Feature filtering. BAG: Bagging. BN: Bayes network.

5.2 Scalability Analysis of DDM and HDSS

We also obtain five more DDMs for three other ICD-10-CM disease categories. Then, through a comprehensive literature review, we identify related work in five other ICD-10-CM categories, for which datasets are not publicly available, but the DDM based HDSS framework would be applicable. Then, we point to ICD-10-CM categories where future work can be explored.

In addition to arrhythmia, we obtain DDMs for type-2 diabetes, breast cancer, urinary bladder disorder, renal pelvis origin nephritis, and hypothyroid disease using the publicly available UCI datasets [37]. We generate DDMs for urinary bladder disorder and renal pelvis origin nephritis from the same acute inflammation dataset. Table 5 shows the number of data instances, features, and classes in these datasets.

In the feature indexing process for these datasets, Tier-2 tests include triceps skin status checking for type-2 diabetes, and physician lesion observations for urinary bladder disorder, renal pelvis origin nephritis, and hypothyroid disease. Tier-3 tests include oral insulin reaction test for type-2 diabetes, cancer cell microscope test for breast cancer, and blood thyroxine test for hypothyroid disease. All other features acquirable without Tier-2 and Tier-3 tests are indexed with 1, including demographic information, historic disease records, body feelings, ECG, and body temperature measurements. Note that basic body feelings, such as existence of lumbar pain or consistent feelings of pushing urine, are considered Tier-1 features. These features can be transferred to a smartphone by the user through a simple user interface. Thus, they are assumed to be available to the PHDS.

The performance results for the generated DDMs are shown in Table 6. In this table, a vertical column represents a DDM for the disease specified at the top. The selected decision maker is shown in the rows indexed with Type. Performance objectives, shown in rows indexed with Obj., vary from binary classifications (B) at Tier-1 to multi-class classification of k classes (M-k) at Tier-2 and Tier-3. The other rows represent corresponding performance metrics, which are stored in performance matrices. A value close to 100 percent or 1 is ideal for all these measurements, except for FPR where a value close to 0 percent is ideal. Results from relevant work are shown in the parentheses next to the corresponding metrics of DDMs for comparison [41], [42], [43], [44]. Sub-optimal performances, where future improvements can be made, are highlighted in the table. The percentages shown below DM1 ACCs, indexed as Impr., show the percentage improvement of the machine learning approach relative to the conventional multi-threshold approach.

TABLE 5
UCI Biomedical Datasets for DDM Generation

Dataset	#Instances	#Features Tier-1	#Features Tier-2	#Features Tier-3	# Classes
Arrhythmia	452	279	279	279	16
Type-2 diabetes	768	5	6	8	2
Accute inflammation	120	5	6	6	2
Hypothyroid disease	3,163	11	13	25	2
Breast cancer	699	0	0	10	2

TABLE 6
Performance Summary of the Generated DDMs Using the Proposed Methods on UCI Biomedical Datasets

	Disease → DDM ↓	Arrhythmia	Diabetes type-2	Breast cancer	Urinary bladder disorder	Renal pelvis origin nephritis	Hypothyroid
	ICD-10-CM	I49.9	E11-*	C50-*	N32.0	N12	E03.9
DM1	Type	RF+F	NB+F	–	RT	NB	RF
	Obj.	B	B	–	B	B	B
	ACC	85.9%	77.6%	–	99.6%	93.7%	94.8%
	(Impr.)	(+28.6%)	(+13.1%)	–	(+19.3%)	(+3.3%)	(+1.1%)
	TNR	88.8%	87.6%	–	99.7%	91.4%	94.9%
	TPR	80.8%	58.9%	–	99.5%	93.8%	10.4%$^{\nabla}$
	FPR	11.2%	12.4%	–	0.3%	8.6%	5.1%
	F1	0.858	0.769	–	0.996	0.950	0.934
	AUC	0.916	0.829	–	0.996	0.996	0.667
	Kappa	0.713	0.485$^{\nabla}$	–	1.000	0.867	0.088$^{\nabla}$
DM2	Type	BAG(BN)+F	NB+F	–	RF+F	RT	ADA(BFTree)
	Obj.	M-16	B	–	B	B	B
	ACC	77.4%	77.6%	–	100% (100%[41])	99.9%(100%[41])	94.8%
	TNR	85.2%	87.6%	–	100%	99.9%	94.9%
	TPR	77.5%	58.9%	–	100%	99.9%	13.6%$^{\nabla}$
	FPR	14.8%	12.4%	–	0%	0.1%	5.1%
	F1	0.751	0.769	–	1.000	0.999	0.934
	AUC	0.913	0.829	–	1.000	0.999	0.684
	Kappa	0.647	0.485$^{\nabla}$	–	1.000	0.998	0.129$^{\nabla}$
DM3	Type	BAG(BN)+F	DEC(BN)	BAG(BN)+F	–	–	J48+F
	Obj.	M-16	B	B	–	–	B
	ACC	77.4% (78.9%[42])	78.9% (76.5%[43])	97.0% (95.5%[43])	–	–	99.3%
	TNR	85.2%	87.2%	98.3%	–	–	99.4% (97.4%[44])
	TPR	77.5%	64.4%	96.6%	–	–	92.4% (97.0%[44])
	FPR	14.8%	12.8%	1.7%	–	–	0.6%
	F1	0.751	0.793	0.976	–	–	0.993 (0.929[44])
	AUC	0.913	0.840 (0.834[43])	0.993 (0.821[43])	–	–	0.957 (0.722[44])
	Kappa	0.647	0.511$^{\nabla}$	0.936	–	–	0.926 (0.926[44])

–Impr.: Improvement over the multi-threshold approach; +F: feature filtering; $^{\nabla}$: sub-optimal performances; DMx: decision maker at Tier-x. Abbreviations for MLMs and performance parameters are from Tables 1 and 4, respectively.

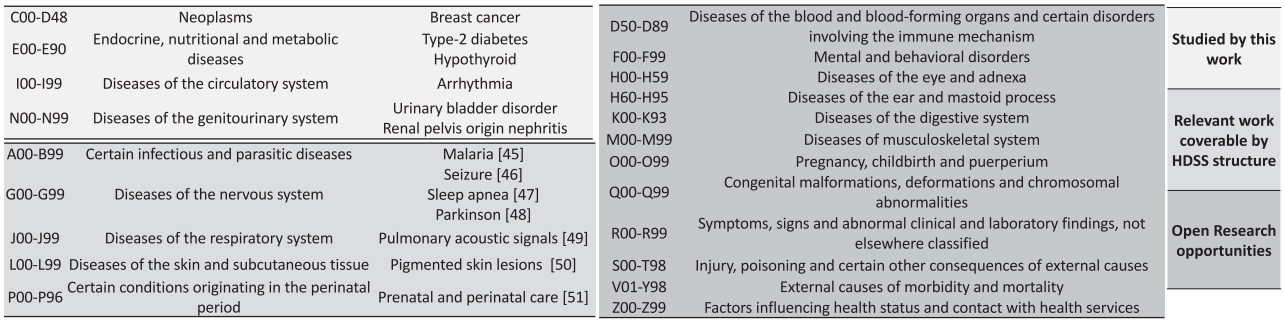


Fig. 8. Coverage of HDSS on all the ICD-10-CM disease categories based on the generated DDMs and analysis of related work.

The quality of training tables, inherited from their original biomedical datasets, plays a significant role in HDSS. A training table with high feature indices may fail to generate decision makers for lower tiers, where the available feature may be null. This limitation arises from the fact that biomedical datasets are often not comprehensive. For example, the breast cancer dataset only contains cell-level features that are only available in Tier-3. Thus, we can only generate DM3 for breast cancer, not DM1 and DM2. If a more comprehensive dataset were available, the DDM

generation procedure could generate all three decision makers, as demonstrated in the cases of arrhythmia, type-2 diabetes, and hypothyroid disease. Hence, to fully utilize the power of HDSS, biomedical datasets need to be prepared carefully, updated consistently, and maintained systematically.

Diagnosis for some diseases needs Tier-3 laboratory measurements. Urinary bladder disorder and renal pelvis origin nephritis can be accurately diagnosed within the first two tiers, whereas diseases like hypothyroid rely heavily on clinical laboratory tests. For hypothyroid disease, DM1 and DM2 have very low TPRs. However, these rates increase dramatically with the inclusion of the Tier-3 blood thyroxine test, indicating the importance of this clinical test in its diagnostic process.

To further evaluate the scalability of HDSS on datasets that are not publicly available yet, we conducted a review of literature published over the last 10-year span. We focused on seven representative works spanning five disease categories that study the effectiveness of MLMs for disease diagnosis and treatment [45], [46], [47], [48], [49], [50], [51]. We present the scope of HDSS in Fig. 8. The figure has three major sections. The lightest section includes the four ICD-10-CM categories covered by the DDMs we generated. The light grey section highlights the categories that contain at least one relevant work with verified MLMs but on private datasets. Thus, if these datasets were made publicly available, HDSS could be applied to these categories as well. Finally, in the dark grey section, we list the open categories where MLMs and biomedical datasets are sparse. They offer opportunities for broadening the scope of HDSS even further.

5.3 Storage Analysis of HDSS

Given the need to store a DDM for each of the 69,000 diseases, an important consideration for establishing the feasibility of HDSS is its tier-wise storage requirement, especially for Tier-1 PHDS that is based on WMSs. In this tier, PHDS may be stored in a cloud server, personal computer base station or ultimately smartphones for end-user convenience. In all these platforms, use of a moderate amount of storage space would be preferred.

To estimate the storage requirement of the HDSS over all diseases, we first generate various base learners under the WEKA-3-7-13 experimental environment for the arrhythmia dataset, which contains 279 features and 400 training instances. With fewer features, base learners require less storage space, which is discussed later. For making a conservative estimate, we assume that a feature count of 279, which is the worst case among our datasets, is the average feature count for the 69,000 diseases. Then, we study the changes required in storage size when varying the training table sizes and ensemble methods, and using the feature-filtering technique.

TABLE 7
Tier-Wise DDM Storage Requirements for Base Learners for 69,000 Diseases Assuming an Average Training Size of 400 Data Instances

MLM type	Storage (GB)
Random tree	2.53
J48	2.66
Naive Bayes	4.46
Decision table	5.46
Bayes network	29.10
Random forest	56.27
k-nearest neighbor	75.01
SVM	75.01
Multilayer perceptron	121.29
Best-first decision tree	949.79

We start HDSS storage analysis with 10 common base learners. For each base learner type, the cumulative storage requirement for 69,000 diseases is given in the corresponding row in Table 7. We include random tree and random forest in this analysis since they are independent of an external base learner. The storage requirement can be seen to vary greatly across different MLM types. Random tree, J48, Naive Bayes, and decision table require far less storage than Bayes network, random forest, k-nearest neighbor ($k = 4$), SVM, multi-layer perceptron, and best-first decision tree. Their storage requirements fall in a moderate range except for best-first decision tree, which needs nearly 1 TB of storage. An increase in the number of training instances may incur some storage overheads, as shown in Fig. 9a. It can be seen that random forest, k-nearest neighbor, SVM, and best-first decision tree are more sensitive to increasing training sizes than the others.

Next, we study storage requirements under performance enhancement techniques, such as ensemble methods and feature filtering. As shown in Fig. 9b, ensemble methods may incur high storage overheads, by as much as 26.8×. This overhead arises from the need to store multiple base learners for a single meta learner. On the other hand, feature filtering enhances the performance at a reduced storage cost, as shown in Fig. 9c. The storage reduction is as much as 32.9× for multilayer perceptron. When both enhancement techniques are used simultaneously, the ratio of the meta learner size and base learner size is shown in Table 8. A value smaller (greater) than 1 represents a storage reduction (increase). Except for J48 which needs 12.4× the original storage, these changes are generally moderate due to the counteracting pressure from ensemble methods and feature filtering. Therefore, we use single base learners for storage estimation of HDSS.

We estimate the tier-wise HDSS storage requirement over all 69,000 diseases by weighting the base learner storage requirements based on what fraction of time that base learner is used in a 10-year survey of data mining models in

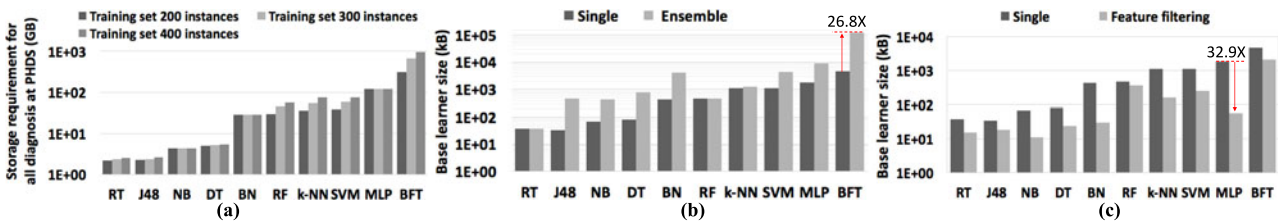


Fig. 9. Storage changes with (a) increasing training instances, (b) ensemble methods, and (c) feature-filtering technique. MLM abbreviations follow Table 1. Single stands for a base learner and does not employ ensemble methods and feature filtering.

TABLE 8
Storage Changes against Base Learners with Both Ensemble Method and Feature Filtering

MLM type	Meta learner size/base learner size
Multilayer perceptron	0.17
k-nearest neighbor	0.22
Random tree	0.39
Bayes network	0.53
Random forest	0.76
Naive Bayes	0.89
SVM	1.45
Decision table	2.80
Best-first decision tree	3.61
J48	12.40

the healthcare domain [52]. This yields the following weights: random forest: 44.8 percent, multilayer perceptron: 17.2 percent, SVM: 17.2 percent, decision table: 7.0 percent, Naive Bayes: 6.9 percent, Bayes network: 6.9 percent, and all other base learners weighted zero. Assuming an average training size of 400 instances, the HDSS storage requirement over all 69,000 diseases was estimated to be 61.75 GB (using the storage values for the base learners shown in Table 7). Such a storage requirement is completely acceptable in today's cloud server or base station oriented BANs.

6 Limitations and Future Work

Currently, we only show HDSS's potential for tackling a disease category by targeting one or few diseases in that category. This is still a far cry from generating a unique disease signature for each of the 69,000 diseases, which is the ultimate goal of HDSS. However, we hope that this work may encourage clinics/hospitals/researchers to start collecting WMS data from individuals to tackle more challenging tasks, such as psychiatric diagnosis. As more models are incorporated into HDSS, its effectiveness as a platform for future personalized diagnosis and medication will improve.

7 Conclusion

In this paper, we described an HDSS that combines WMSs and CDSSs. It incorporates a hierarchical and closed-loop multi-tier structure supported by robust machine learning ensembles. We proposed an automatic DDM generation procedure that can monitor various diseases in parallel. We demonstrated the feasibility of HDSS by generating six DDMs for diseases drawn from four ICD-10-CM categories. We showed that significant disease classification accuracy can be obtained through physiological data obtained from WMSs: arrhythmia (86 percent), type-2 diabetes (78 percent), urinary bladder disorder (99 percent), renal pelvis nephritis (94 percent), and hypothyroid (95 percent). Then, we discussed how HDSS can also be applied to other disease categories when datasets for those diseases become available. We estimated that the DDMs for all reported human diseases need around 62 GB of storage in the WMSs tier. This is well within the means of current technology.

References

[1] J. A. Salomon, et al., "Healthy life expectancy for 187 countries, 1990-2010: A systematic analysis for the global burden disease study 2010," *Lancet*, vol. 380, no. 9859, pp. 2144–2162, 2013.

[2] Health care costs 101, 2016. [Online]. Available: https://www.chcf.org/publications/2016/12/health-care-costs-101

[3] M. A. Makary and M. Daniel, "Medical error-the third leading cause of death in the US," *British Med. J.*, vol. 353, 2016, Art. no. i2139.

[4] D. W. Bates, M. Cohen, L. L. Leape, J. M. Overhage, M. M. Shabot, and T. Sheridan, "Reducing the frequency of errors in medicine using information technology," *J. Amer. Med. Informat. Assoc.*, vol. 8, no. 4, pp. 299–308, 2001.

[5] D. L. Hunt, R. B. Haynes, S. E. Hanna, and K. Smith, "Effects of computer-based clinical decision support systems on physician performance and patient outcomes: A systematic review," *J. Amer. Med. Assoc.*, vol. 280, no. 15, pp. 1339–1346, 1998.

[6] J. Grytten and R. Sorensen, "Practice variation and physician-specific effects," *J. Health Econ.*, vol. 22, no. 3, pp. 403–418, 2003.

[7] M. M. Baig and H. Gholamhosseini, "Smart health monitoring systems: An overview of design and modeling," *J. Med. Syst.*, vol. 37, no. 2, pp. 1–14, 2013.

[8] H. Lee, et al., "A graphene-based electrochemical device with thermoresponsive microneedles for diabetes monitoring and therapy," *Nature Nanotechnol.*, vol. 11, no. 6, pp. 556–572, 2016.

[9] U. Anliker, et al., "AMON: A wearable multiparameter medical monitoring and alert system," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 4, pp. 415–427, Dec. 2004.

[10] G. Lopez, V. Custodio, and J. I. Moreno, "LOBIN: E-Textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 6, pp. 1446–1458, Nov. 2010.

[11] L. Pollonini, N. O. Rajan, S. Xu, S. Madala, and C. C. Dacso, "A novel handheld device for use in remote patient monitoring of heart failure patients–Design and preliminary validation on healthy subjects," *J. Med. Syst.*, vol. 36, no. 2, pp. 653–659, 2012.

[12] P. Pandian, et al., "Smart vest: Wearable multi-parameter remote physiological monitoring system," *Med. Eng. Physics*, vol. 30, no. 4, pp. 466–477, 2008.

[13] Health watch medical grade smart clothing technology, 2016. [Online]. Available: http://www.personal-healthwatch.com/technology

[14] G. N. Forrest, T. C. van Schooneveld, R. Kullar, L. T. Schulz, P. Duong, and M. Postelnick, "Use of electronic health records and clinical decision support systems for antimicrobial stewardship," *Clinical Infectious Diseases*, vol. 59, pp. 122–133, 2014.

[15] H. Quan, et al., "Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data," *Med. Care*, vol. 43, pp. 1130–1139, 2005.

[16] Growth trends, consumer attitudes, and why smartwatches will dominate, 2015. [Online]. Available: http://www.businessinsider.com/the-wearable-computing-market-report-2014–10

[17] T. Majumder, X. Li, P. Bogdan, and P. Pande, "NoC-enabled multi-core architectures for stochastic analysis of biomolecular reactions," in *Proc. IEEE Des. Autom. Test Europe Conf.*, 2015, pp. 1102–1107.

[18] P. Kulkarni and Y. Öztürk, "Requirements and design spaces of mobile medical care," *Mobile Comput. Commun. Rev.*, vol. 11, no. 3, pp. 12–30, 2007.

[19] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An ad-hoc sensor network infrastructure for emergency medical care," in *Proc. Int. Workshop Wearable Implantable Body Sensor Netw.*, (2004). [Online]. Available: http://www.eecs.harvard.edu/~mdw/papers/codeblue-bsn04.pdf

[20] K. Wac, A. van Halteren, and D. Konstantas, "QoS-predictions service: Infrastructural support for proactive QoS-and context-aware mobile services (position paper)," in *On the Move to Meaningful Internet System*. Berlin, Germany: Springer, 2006, pp. 1924–1933.

[21] R. G. Lee, K. C. Chen, C. C. Hsiao, and C. L. Tseng, "A mobile care system with alert mechanism," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 5, pp. 507–517, Sep. 2007.

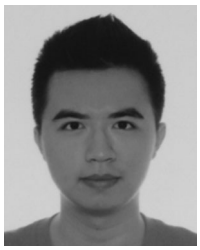
[22] D. Apiletti, E. Baralis, G. Bruno, and T. Cerquitelli, "Real-time analysis of physiological data to support medical applications," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 3, pp. 313–321, May 2009.

[23] A. L. Goldberger, L. A. N. Amaral, J. M. Hausdorff, P. C. Ivanov, C.-K. Peng, and H. E. Stanley, "Fractal dynamics in physiology: Alterations with disease and aging," *Proc. Nat. Academy Sci. United States America*, vol. 99, pp. 2466–2472, 2002.

[24] P. Bogdan, S. Jain, and R. Marculescu, "Pacemaker control of heart rate variability: A cyber physical system perspective," *ACM Trans. Embedded Comput. Syst.*, vol. 12, pp. 1–22, Mar. 2013.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[26] P. Melville and R. J. Mooney, "Constructing diverse classifier ensembles using artificial training examples," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, vol. 3, pp. 505–510.
[27] S. Youm, G. Lee, S. Park, and W. Zhu, "Development of remote healthcare system for measuring and promoting healthy lifestyle," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2828–2834, 2011.
[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[29] TensorFlow, 2016. [Online]. Available: https://www.tensorflow.org
[30] A. M. Nia, M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Energy-efficient long-term continuous personal health monitoring," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 85–98, Apr.–Jun. 2015.
[31] I. K. Dimov, L. Basabe-Desmonts, J. L. Garcia-Cordero, B. M. Ross, A. J. Ricco, and L. P. Lee, "Stand-alone self-powered integrated microfluidic blood analysis system (SIMBAS)," *Lab Chip*, vol. 11, no. 5, pp. 845–850, 2011.
[32] R. C. Lozoya, "Radiofrequency ablation planning for cardiac arrhythmia treatment using biophysical modeling and machine learning approaches," Ph.D. dissertation, the IT Doctoral School, Universite Nice Sophia Antipolis, Nice, France, Sept. 2015.
[33] P. Bogdan, "A cyber-physical systems approach to personalized medicine: Challenges and opportunities for NOC-based multicore platforms," in *Proc. IEEE Des. Autom. Test Europe Conf.*, 2015, pp. 253–258.
[34] C.-L. Chi, W. N. Street, J. G. Robinson, and M. A. Crawford, "Individualized patient-centered lifestyle recommendations: An expert system for communicating patient specific cardiovascular risk information and prioritizing lifestyle options," *J. Biomed. Informat.*, vol. 45, no. 6, pp. 1164–1174, 2012.
[35] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
[36] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 6, pp. 1893–1905, Nov. 2015.
[37] UCI machine learning repository, 2016. [Online]. Available: http://archive.ics.uci.edu/ml
[38] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature Extraction*. Berlin, Germany: Springer, 2006, pp. 1–25.
[39] M. P. Tarvainen, J.-P. Niskanen, J. A. Lipponen, P. O. Ranta-Aho, and P. A. Karjalainen, "Kubios HRV–heart rate variability analysis software," *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 210–220, Jan. 2014.
[40] B. H. Dobkin and A. Dorsch, "The promise of mHealth: Daily activity monitoring and outcome assessments by wearable sensors," *Neurorehabilitation Neural Repair*, vol. 25, no. 9, pp. 788–798, 2011.
[41] M. Arif and S. Basalamah, "Similarity-dissimilarity plot for high dimensional data of different attribute types in biomedical datasets," *Int. J. Innovative Comput. Inf. Control*, vol. 8, no. 2, pp. 1275–1297, 2012.
[42] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, "Modular neural network based arrhythmia classification system using ECG signal data," *Int. J. Inf. Technol. Knowl. Manage.*, vol. 4, no. 1, pp. 205–209, 2011.
[43] X. H. Cao, I. Stojkovic, and Z. Obradovic, "A robust data scaling algorithm to improve classification accuracies in biomedical data," *BMC Bioinf.*, vol. 17, pp. 1–10, 2016.
[44] D. Suryakumar, A. H. Sung, and Q. Liu, "Influence of machine learning versus ranking algorithm on the critical dimension," *Int. J. Future Comput. Commun.*, vol. 2, no. 3, pp. 215–220, 2013.
[45] D. K. Das, M. Ghosh, M. Pal, A. K. Maiti, and C. Chakraborty, "Machine learning approach for automated screening of malaria parasite using light microscopic images," *Micron*, vol. 45, pp. 97–106, 2013.
[46] M. Shoaib, N. K. Jha, and N. Verma, "Signal processing with direct computations on compressively sensed data," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 1, pp. 30–43, Jan. 2015.
[47] A. H. Khandoker, M. Palaniswami, and C. K. Karmakar, "Support vector machines for automated recognition of obstructive sleep apnea syndrome from ECG recordings," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 1, pp. 37–48, Jan. 2009.
[48] N. M. Tahir and H. H. Manap, "Parkinson disease gait classification based on machine learning approach," *J. Appl. Sci.*, vol. 12, no. 2, pp. 180–185, 2012.
[49] R. Palaniappan, K. Sundaraj, and S. Sundaraj, "A comparative study of the SVM and k-NN machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals," *BMC Bioinf.*, vol. 15, no. 1, pp. 1–8, 2014.
[50] K. Korotkov and R. Garcia, "Computerized analysis of pigmented skin lesions: A review," *Artif. Intell. Med.*, vol. 56, no. 2, pp. 69–90, 2012.
[51] F. R. Cerqueira, et al., "NICeSim: An open-source simulator based on machine learning techniques to support medical research on prenatal and perinatal care decision making," *Artif. Intell. Med.*, vol. 62, no. 3, pp. 193–201, 2014.
[52] I. Yoo, et al., "Data mining in healthcare and biomedicine: A survey of the literature," *J. Med. Syst.*, vol. 36, no. 4, pp. 2431–2448, 2012.



Hongxu Yin (S'15) received the BEng degree in electrical and electronics engineering from Nanyang Technological University, Singapore, in 2015. He is currently working toward the PhD degree in the Electrical Engineering Department, Princeton University. His research interests include machine learning, artificial intelligence, wearable medical sensors, Internet-of-things, secure computing, and low-power analog and digital systems. He is a student member of the IEEE.

Niraj K. Jha (S'85-M'85-SM'93-F'98) received the BTech degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, India in 1981, the MS degree in electrical engineering from S.U.N.Y. at Stony Brook, New York, in 1982, and the PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign, Illinois, in 1985. He is a professor of electrical engineering with Princeton University. He received the Distinguished Alumnus Award from I.I.T., Kharagpur, in 2014. He received the AT&T Foundation Award and NEC Preceptorship Award for research excellence, NCR Award for teaching excellence, six School of Engineering and Applied Science Teaching Commendations, and Princeton University Graduate Mentoring Award. He has served as the editor-in-chief of the *IEEE Transactions on VLSI Systems* and an associate editor of the *IEEE Transactions on Circuits and Systems I and II*, the *IEEE Transactions on VLSI Systems*, the *IEEE Transactions on Computer-Aided Design*, the *IEEE Transactions on Computers*, the *Journal of Electronic Testing: Theory and Applications*, and the *Journal of Nanotechnology*. He is currently serving as an associate editor of the *IEEE Transactions on Multi-Scale Computing Systems* and the *Journal of Low Power Electronics*. He has served as the program chairman of the 1992 Workshop on Fault-Tolerant Parallel and Distributed Systems, the 2004 International Conference on Embedded and Ubiquitous Computing, and the 2010 International Conference on VLSI Design, as the director of the Center for Embedded System-on-a-Chip Design funded by New Jersey Commission on Science and Technology, as the associate director of the Andlinger Center on Energy and the Environment, and on the program committees of more than 150 conferences and workshops. He has co-authored or co-edited five books titled *Testing and Reliable Design of CMOS Circuits* (Kluwer, 1990), *High-Level Power Analysis and Optimization* (Kluwer, 1998), *Testing of Digital Systems* (Cambridge University Press, 2003), *Switching and Finite Automata Theory*, 3rd edition (Cambridge University Press, 2009), and *Nanoelectronic Circuit Design* (Springer, 2010). He has also authored 15 book chapters. He has authored or co-authored more than 430 technical papers. He has coauthored 14 papers, which have won various awards. These include the Best Paper Award at ICCD'93, FTCS'97, ICVLSID'98, DAC'99, PDCS'02, ICVLSID'03, CODES'06, ICCD'09, and CLOUD'10. A paper of his was selected for "The Best of ICCAD: A collection of the best IEEE International Conference on Computer-Aided Design papers of the past 20 years," two papers by IEEE Micro Magazine as one of the top picks from the 2005 and 2007 Computer Architecture conferences, and two others as being among the most influential papers of the last 10 years at IEEE Design Automation and Test in Europe Conference. He has co-authored another six papers that have been nominated for best paper awards. He has received 14 U.S. patents. His research interests include embedded computing, secure computing, machine learning, monolithic 3D IC design, low power hardware/software design, and computer-aided design of integrated circuits and systems. He has given several keynote speeches in the areas of nanoelectronic design/test and embedded systems. He is a fellow of the IEEE and the ACM.