Journal of Medical Systems (2020) 44: 76
https://doi.org/10.1007/s10916-019-1518-8

MOBILE & WIRELESS HEALTH

A Scalable Smartwatch-Based Medication Intake Detection System Using Distributed Machine Learning

Donya Fozoonmayeh[1] · Hai Vu Le[1] · Ekaterina Wittfoth[1] · Chong Geng[1] · Natalie Ha[1] · Jingjue Wang[1] · Maria Vasilenko[1] · Yewon Ahn[2] · Diane Myung-kyung Woodbridge[1]

Received: 14 July 2019 / Accepted: 12 December 2019 / Published online: 28 February 2020

Abstract

Poor Medication adherence causes significant economic impact resulting in hospital readmission, hospital visits and other healthcare costs. The authors developed a smartwatch application and a cloud based data pipeline for developing a user-friendly medication intake monitoring system that can contribute to improving medication adherence. The developed Android smartwatch application collects activity sensor data using accelerometer and gyroscope. The cloud-based data pipeline includes distributed data storage, distributed database management system and distributed computing frameworks in order to build a machine learning model which identifies activity types using sensor data. With the proposed sensor data extraction, preprocessing and machine learning algorithms, this study successfully achieved a high F1 score of 0.977 with 13.313 seconds of training time and 0.139 seconds for testing.

Keywords Medication adherence · Internet of things · Wearable · Smartwatch · Health monitoring · Machine learning · Cloud computing · Distributed information systems · Distributed computing · Distributed databases.

Introduction

In the United States, over 117 million people have more than one chronic diseases that often require medication [8]. Medication adherence measures how closely patients follow their prescribed treatment regimens including dosage and time [38]. Unfortunately, the medication adherence rate for patients with chronic diseases is only about 50% which is much lower than the adherence rate for patients with acute diseases, showing gradual drops in their first few months of clinical trials [23].

Medication non-adherence costs $100 billion every year in the United States, causing hospital readmission, emergency department and physician visits, death, and other healthcare costs [38]. The high costs could get worse as outpatient medication expenditure increases by over 10% per year, with increases in the aging population and patients with chronic diseases [46]. Therefore, increased medication adherence can help control symptoms and potentially reduce overall medical cost.

The two main factors causing medication non-adherence are patient's stress and the complexity of the tasks [50, 53]. First, a patient's emotional and physical stress is the main factor causing medication non-adherence [38, 43]. Emotional stress affecting medication non-adherence includes depression, denial or anger about the illness and fear of medication addiction and its side effects. Physical stress factors include illness and cognitive and physical declines. Second, The complexity of medication intake includes the number of medications to take, the frequency, treatment cost, and medication refill policy and procedure. Both stress and complexity affect patients' motivation, which is the most critical factor for long-term medication adherence [36]. While stress is hard to be controlled by external factors, the complexity of medication adherence could be improved with the help of technology.

In order to develop an effective medication intake monitoring system which can contribute to improving medication adherence [2], it is critical to consider social acceptance, ease of use, and time and cost efficiency for

This article belongs to the Topical Collection *Mobile & Wireless Health*

Donya Fozoonmayeh, Hai Vu Le and Ekaterina Wittfoth have contributed equally.

✉ Diane Myung-kyung Woodbridge
dwoodbridge@usfca.edu

Extended author information available on the last page of the article.

enhancing user experience. Developing a user-friendly real-time medication intake monitoring system can simplify medication intake process by detecting medication intake activities and tracking the activities [33]. Many research studies suggest that perceived ease of use, usefulness, and benefits are closely related to a user's acceptance, satisfaction and intention to use a mobile health monitoring system which may directly affect medication adherence [28, 55]. Adopting a lightweight wearable device with convenient and efficient user interface (UI) can improve usability for monitoring a patient's medication intake activities and provide reminders and feedback on time. Cost and time efficiency is also critical for patient satisfaction and adherence rate [24, 44, 47]. The use of Internet of Things (IoT) health monitoring solutions can reduce 68.3% of the healthcare cost by lowering hospitalization rate and physician office visits; although initial costs of device and service could be an obstacle [45]. For reducing costs caused by initial hardware design, development, server and infrastructure maintenance, adopting off-the-shelf IoT wearable devices and cloud services could contribute to cost reduction [48]. Many IoT solutions utilize various computing infrastructure including cloud computing and edge computing for improving time and cost efficiency and reducing delay for acquiring, storing and processing data by efficiently organizing and distributing data [40]. These computing tools provide a seamless interaction between a server and a device and allow a user to receive a timely feedback to prevent any health-related adverse events.

In this research, we focused on a low-cost real-time medication intake monitoring system, by designing and developing a smartwatch application and utilizing distributed data storage and distributed machine learning models. The smartwatch application collects activity data from a user and sends data to a distributed data storage, Amazon Web Services (AWS, [1]) Simple Storage Service (S3, [3]). Preprocessed data is stored in a distributed database, MongoDB [35] that is connected to a distributed processing framework, Apache Spark [4]. We utilized off-the-shelf devices and cloud services in order to provide service at low cost as well as with stability.

The rest of this paper is organized as follows: Section "Related work" covers existing medication intake monitoring procedures and systems. Sections "System architecture" and "Algorithms" contain a system architecture and algorithm details. Section "Experiment results" contains experiment design, specifications of different computing settings and experiment results under different machine learning algorithms. Section "Conclusion" provides conclusion and future work.

Related work

Medication intake monitoring approaches fall under two broad categories: direct and indirect. Direct methods include direct observation of a patient taking medication, laboratory detection of drug in a patient's biological fluids or in biomarkers. Indirect methods are represented by a patient's reporting, pill counting, medication refill history tracking, and electric tracking systems using cameras or wearables. While direct methods are most accurate in monitoring medication adherence, they are most costly, invasive and time-consuming [25]. Indirect methods, in contrast, provide relatively inexpensive and effective tools to monitor medication adherence. As cost and ease of use determines successful medication adherence, in this section we discuss various indirect methods.

Conventionally, patients record and follow their medication intake using medication log sheets, text message reminders or smartphone logging applications [30, 39, 49]. Self-reporting methods including log sheets and smartphone logging applications require users to answer questions of whether she or he had taken medications on schedule [20]. An electronic pill box or image scanning system could also track a user's medication intake behavior [17, 22]. However, unfortunately, user's cognitive impairment or age related memory loss, busy schedule and medical symptoms could affect the accuracy of the reporting outcome [42]. As the number of requested tasks is highly related to task complexity and adherence rate [29], minimizing a user's manual inputs, such as opening an application or pressing buttons, and seamlessly detecting medication intake is critical.

In order to improve the medication adherence rate, many recent studies have developed systems that utilize low-cost sensors, which can record a series of activities during medication intake and provide feedback by analyzing sensor readings. By automatically recognizing medication intakes among others activities using data collected via sensors, these system could detect whether a patient has taken their medication during a desirable time window and hence could be used to provide reminders in case of missed medication intakes. A seamless integration of such systems into patients' lifestyle, for example in form of a mobile app, can leverage timely and natural interaction with patients, requiring minimal changes in their habits or daily routines, and thus, promising an improvement in their medication adherence. Sensor-based systems can utilize both or either wearable and non-wearable sensors to monitor user behavior and activities. Non-wearable systems generally utilize sensors capturing images and videos, while wearable devices utilize activity sensors including accelerometer and gyroscope which collect 3-dimensional acceleration, orientation and angular velocity. Hasanuzzaman's work used radio-frequency identification (RFID) tags attached to a medication bottle along with captured images from a video camera to a subject's face and activities [21]. Tucker, et al. developed data mining driven methodology, which utilizes Microsoft Kinect sensors, to model and predict patients' adherence to medication protocols, based on variations in their motions [52]. While non-wearable solutions are low-cost and do not require additional effort from patients like wearing a device, their use is still restricted to a certain area such as a patient's house and often raise privacy concerns. Chen's study utilizes inertial sensors and an RGB-Depth camera in addition to an accelerometer and gyroscope that is attached to a patient's wrist to collect data, to which dynamic time-warping is applied to measure the similarity between time-series data with different lengths [12]. Kalantarian's research employs smartwatches attached to a patient's both wrists for collecting and processing accelerometer and gyroscope data in order to detect a series of activities including opening a bottle and twisting a cap by using the distribution of the sensor readings [30]. The study requires the patient to wear sensors on both wrists and only applied one classification algorithm (namely, decision trees). To address that issue, Kalantarian extended his study to offer a system and algorithms based on data collected from a smart necklace. The system offers opportunities to detect whether the medication has been ingested based on the skin movement in the lower part of the neck during a swallow using a piezoelectric sensor [31]. The system applies Bayesian networks to classify between chewable vitamins, saliva swallows, medication capsules, speaking, and drinking water and was able to reach the average precision and recall of 90.17 % and 88.9 %, respectively. Yet, wearing a necklace might be uncomfortable for patients, thus lowering the usability and system acceptance rate.

Considering the ease of use, it is better to use embedded sensors in one device which is easy and light to wear. Additionally, a device that supports seamless data transfer, has a long battery life and is of durable quality improves usability. In that sense, a smartwatch provides higher usability and social acceptance along with the capabilities of measuring and transferring activity data. A survey with 221 people from Kalantarian's work shows that 72% of participants responded positively to wearing smartwatches [30].

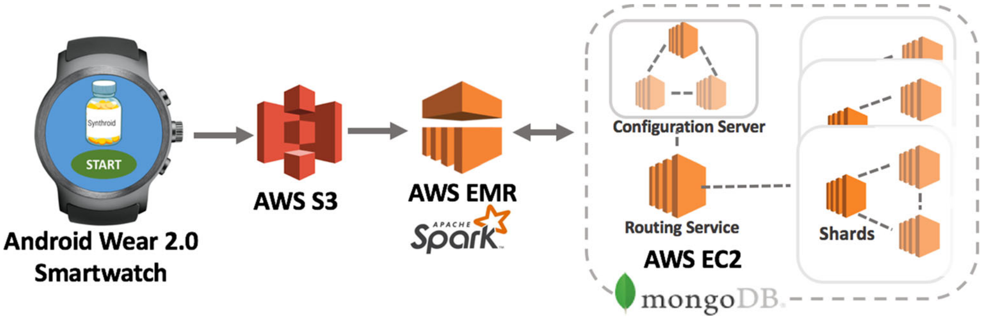
System architecture

In order to develop a low-cost, scalable, reliable and time-efficient medication intake monitoring framework, we utilized a smartwatch (supporting various embedded activity sensors along with a cellular connection), distributed data storage and processing engines. In this study, activity sensor readings for different types of activities are transferred from a smartwatch to a cloud storage. Then, the system processes and transforms raw data into a DataFrame that is structured data with columns and rows of statistical descriptive features using a distributed processing engine and stores it in a distributed schemaless database. In order to develop a machine learning model with high accuracy and efficiency from a large volume of high-frequency data, we applied and validated multiple machine learning algorithms written in the distributed processing framework. Figure 1 shows the designed and developed data science pipeline.

Mobile application

Smartwatches are effective activity monitoring devices because they already contain embedded sensors that can capture a wide range of movements. For example, smartwatches contain a three-axis accelerometer, gyroscope, near-field communication (NFC), and heart rate monitor. These seamlessly integrated sensors provide a much less obtrusive monitoring experience in comparison to smartphones or other wearable devices such as a heart rate monitor chest strap. Sensor data collected from a smartwatch

Fig. 1 Data science pipeline



application plays a critical role in providing contextual information which can be used for analyzing user behavior and generating relevant feedback for patients. Additionally, information provided from a smartwatch is more easily accessible than information provided from other devices including a laptop, tablet or smartphone, because of its compactness and adjacency to the user [26]. In this study, we utilized an LG Watch Sport - the first Android watch running on Android Wear OS 2.0 which provides improved user interface and a cellular connectivity [6]. The list of available biosensors that LG Watch Sport supports is listed in Table 1. As LG Watch Sport supports cellular connection, collected sensor data can be directly transmitted to the cloud storage without being synchronized to a smartphone or without WiFi connectivity.

In this study, we collected 3-axis accelerometer and gyroscope data with a sensor delay of up to 5 milliseconds. These two sensors play a critical role in detecting activity types – the accelerometer sensor measures acceleration while the gyroscope measures orientation and angular velocity of activity. In order to save storage space on the device and reduce the amount of data transferred over the network, the system collects data only when there is a change in sensor readings.

Cloud services

Accelerometer and gyroscope sensors embedded in the smartwatch collect three-dimensional data with a frequency of 200 Hz. This multidimensional high-frequency time-series data requires scalable solutions for data storage, database system, data preprocessing, and machine learning model development. Cloud computing utilizes storage and computing resources located in multiple data centers connected via a network, and provides services on demand. Cloud computing is highly scalable and user-friendly, reacting to user needs dynamically by scaling resources, and providing IT infrastructure and maintenance services. Allowing resources and services to be shared by multiple users, cloud computing minimizes cost and became an economic and powerful tool [10, 13, 18, 57]. Therefore, a cloud service which is scalable and accessible could be the best solution for storing and processing the high-frequency sensor data in the multi-user setting. Since motion data is captured with millisecond granularity, the size of the data increases exponentially. Acknowledging these constraints, we identified AWS as a platform that provides cost-effective storage and computing frameworks [1].

Table 1 A list of biosensors embedded in LG Watch Sport and monitored attributes

Sensor	Monitored attributes
Gyroscope	Rotation
Accelerometer	Acceleration
Photoplethysmogram (PPG)	Heart rate
Barometer	Atmosphere pressure

Distributed data storage

For storing raw sensor data collected from a smartwatch, we utilized networked data stores which support high data availability by replicating data in multiple servers. With AWS Simple Storatge Service (S3), data is accessible from anywhere with an option to replicate data in multiple storage across many regions in the world. Additionally, S3 offers a secure infrastructure through access policy options that allows only authorized users to access the data. AWS S3 also ensures scalability and flexibility by parallelizing requests and allowing any size and type of object, while minimizing time and cost for server maintenance [3].

Distributed database

In the last two decades, tech companies started tracking detailed user behaviors through websites and IoT devices in real-time, which caused a huge volume of data with an evolving schema. For storing IoT data with explosive volume growth, the needs of an affordable but robust system arose. Many of the new database management systems support distributed data sources by dividing and storing data in different servers (shards) and improve data availability by maintaining replicas in multiple servers [11, 16].

MongoDB, one of the most popular distributed databases, stores data in a schemaless JSON document format allowing users to add and remove fields easily. MongoDB is designed to scale out and split up data across multiple servers. MongoDB takes care of loading data across a cluster, balancing data distribution in multiple servers and routing user requests to the server which has the relevant data points. These capabilities allow users to focus on programming rather than low-level system architecture and data distribution [35].

For developing a distributed database, the system utilizes several AWS Elastic Compute Cloud (EC2) instances with MongoDB installed. For developing a distributed database management system, a routing server (mongos), configuration nodes and data shards and their replica nodes are launched (Fig. 1). Mongos service node takes user requests and routes them to the right instance which contains requested data. The configuration nodes include one primary (master) and two secondaries (slaves) and manage metadata of the overall database. We divided the original sensor readings into shard nodes where each shard's primary and secondaries maintain a subset of preprocessed sensor readings. For configuration and data shard nodes, the system maintains one primary and multiple secondary nodes for each shard in case of a primary node failure. Each primary node is in charge of read and write operations and copies data to secondaries. Secondary nodes maintain replicated data which can be used when a master node fails due to networking, power outage, and other system failures.

Distributed computing

Hadoop's MapReduce, introduced in 2004, implemented efficient distributed techniques in an attempt to speed up large scale data analysis [15]. MapReduce splits data into smaller chunks across different nodes, and subsequently maps and processes a task, e.g., filtering and sorting, in parallel. The output of a mapped task becomes the input of a reduce operation, which performs a summary operation. This highly-effective model allows users to design programs with successive Map and Reduce operations, and is a popular and powerful programming paradigm.

Apache Spark adopts the MapReduce model, but executes a task close to 100 times faster than MapReduce by processing data in memory. Also, Spark uses efficient job scheduling and recovery model using directed acyclic graph (DAG) representation, and still runs 10 times faster in disk than MapReduce [5, 19, 56].

For processing sensor data and applying machine learning algorithms using Spark, we utilized AWS Elastic MapReduce (EMR) which uses Hadoop's YARN (Yet Another Resource Negotiator) for provisioning the cluster's hardware resources (EC2 instances) and installs the required software for running Apache Spark (Fig. 2).

Algorithms

In order to process high-frequency sensor data and classify medication intake activities, we designed and developed a preprocessing algorithm to impute missing data and extract statistical features and applied four machine learning algorithms being executed on a Spark cluster.

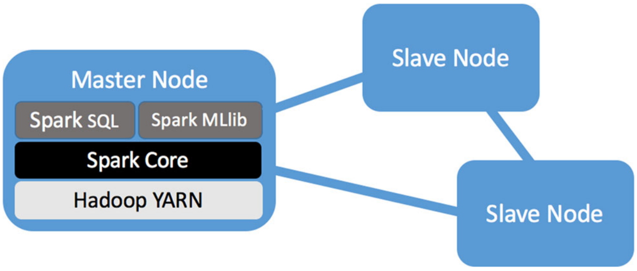


Fig. 2 AWS EMR cluster architecture

Preprocessing algorithm

In order to save storage and computing resources, the data is only collected from the smartwatch application when there is a new sensor event triggered by an accelerometer or a gyroscope. Therefore, for discretizing the data and calculating the statistics of data, missing data imputation was necessary. Additionally, as this work applies classification algorithms to different lengths of time-series data from the 3-axis accelerometer and gyroscope, data discretizion was applied along with feature extraction. The pseudocode for missing data imputation is listed in Algorithm 1.

Algorithm 1 Missing Data Imputation Algorithm Pseudocode.

```
if $time(first\_reading) > min\_timestamp$ then
    $time = min\_timestamp$
else
    $time = time(first\_reading)$
end if
while $time(first\_reading) > time$ do
    $reading = sensor\_reading(first\_reading)$
    $time += time\_increment$
end while
for $data$ in $sensor\_readings$ do
    $prev\_reading = reading$
    while $time < time(data)$ do
        $reading = prev\_reading$
        $timee += time\_increment$
    end while
    $reading = sensor\_reading(data)$
    $time += time\_increment$
end for
if $time <= max\_timestamp$ then
    $reading = prev\_reading$
    $time += time\_increment$
end if
```

Once missing data is imputed, we discretized high-frequency data which was collected every five milliseconds. Since the time duration of each data varies, we reduced the time-series data length of n to the length of f ($f \leq n$) and calculated statistics for the entire data and over each sliding window. When the original time-series after imputing missing data is $C = c_1, ..., c_n$, the mean over the sliding window ($\overline{C}$) is calculated by Eq. 1. In addition to the mean in Eq. 1, we also calculated other aggregate measures including minimum, maximum, 5, 25, 50, 75 and 95 percentiles and standard deviation accordingly for the entire time frame and each sliding window. In addition to the mean, adding statistical values as features help estimate data distribution

Table 2 EC2 instance configurations for MongoDB (Given CPU, memory, storage and price information are for each node)

Role	Number of nodes	Number of CPU	Memory	Storage	Price
Shard 1	3	1	2 GB	8 GB	\$0.023/hr
Shard 2	3	1	2 GB	8 GB	\$0.023/hr
Routing	1	1	1 GB	8 GB	\$0.012/hr
Configuration	3	1	1 GB	8 GB	\$0.012/hr

and outliers. For example, percentile values provide a better understanding about the distribution of the data [9].

$$\overline{\mu}_i = \frac{f}{n} \sum_{j=\frac{n}{f}(i-1)+1}^{\frac{n}{f}i} c_j \tag{1}$$

Machine learning algorithms

In order to accurately classify the medication intake activity, we grouped the activity labels into a binary class — a medication intake activity and not a medication intake activity (including other activities). Using these labels, we applied four different supervised learning algorithms and compared their predictive performance using metrics such as F1 scores, as well as execution time.

Random forest

Random forest is an ensemble-based supervised learning algorithm that aggregates multiple decision trees [41]. The algorithm uses random sampling of training data when building trees and a random subset of features when splitting the nodes. This inherent randomness within the trees avoids overfitting issues complicit with deterministic decision trees, which allows random forest to perform well without much of hyperparameter tuning. Each decision tree in a random forest learns from random samples which are drawn using bootstrapping. Predictions for testing are calculated by averaging the predictions of each decision tree [7].

Gradient-boosted tree

Gradient boosting is an ensemble-based machine learning method that can be used for classification and regression. The principle behind gradient boosting is using an ensemble of weak decision tree stumps to form a strong classifier or regressor. Unlike the random forest algorithm, the gradient boosting algorithm puts more weight on previously misclassified samples when generating successive trees. Just like any other supervised machine learning algorithm, the goal of gradient boosting is to minimize a loss function such as mean squared error (MSE, (2)) or mean absolute error (MAE, (3)) [34].

$$MSE = \frac{1}{n} \sum_{k=1}^{n} (predicted_k - true_k)^2 \tag{2}$$

$$MAE = \frac{1}{n} \sum_{k=1}^{n} \mid predicted_k - true_k \mid \tag{3}$$

Logistic regression

Logistic regression is a widely used statistical supervised machine learning algorithm that predicts the probability that an input value belongs to a particular category by fitting the data to a linear regression model, which is then passed to the logistic function in Eq. 4 [14, 37]. The main strength of logistic regression is the interpretability of the model outputs. The algorithm can also be regularized to avoid overfitting and is often used as a base model for classification problems.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

Support vector machine

Support Vector Machine (SVM) is a machine learning algorithm that classifies class labels by solving a convex optimization problem to find a separating hyperplane, Eq. 5

Table 3 EMR cluster types used for launching Apache Spark (Given CPU, memory, storage and price information are for each node)

Cluster	Role	Number of CPU	Memory	Storage	Price
Cluster 1	Primary	4	8 GB	32 GB	\$0.030/hr
	Secondary (Total 2 Nodes)	4	8 GB	32 GB	\$0.030/hr
Cluster 2	Primary	8	15 GB	80 GB	\$0.070/hr
	Secondary (Total 2 Nodes)	8	7 GB	32 GB	\$0.035/hr

in a Hilbert space that maximizes the margin between the two classes [32].

$$w \cdot x + b = 0 \tag{5}$$

SVM uses a nonlinear function to map vectors in the input space to a higher dimensional space where the classes can be linearly separated [51].

Experiment results

For validating the designed data science pipeline, we deployed the distributed systems for storing and processing sensor data from smartwatches. The experiment setting section describes the details of hardware being used and human subjects along with performed activities. The result section demonstrates the accuracy and time efficiency of the developed system.

Experiment setting

In this study, the system was designed to store a large volume of high frequency sensor data stream, extract features and apply machine learning algorithms with scalability and time efficiency using cloud-based frameworks. The recruited human subjects performed various activities for collecting data using the developed smartwatch application.

System architecture setting

We utilized Amazon Web Services for implementing a cloud-based data pipeline to preprocess, store and apply machine learning algorithms using distributed frameworks. Preprocessed data is stored in MongoDB and the specifications of our launched AWS Elastic Compute Cloud (EC2) instances for MongoDB are in Table 2. For applying machine learning algorithms to data from MongoDB, we used Apache Spark installed on two different AWS Elastic Map Reduce (EMR) clusters where each has one primary and two secondary nodes. The specifications of each EMR are outlined in Table 3.

Subject and data collection

For the experiment, we collected data from 24 individuals listed in Table 4. Each individual performed medication intake activities wearing watches on either their left or right wrists. In addition, individuals performed non-medication intake activities including texting, walking, writing and opening and drinking a bottled water (Table 5). The subjects repeated each activity five times. The data is randomly split into 80% and 20% for training machine learning models and validating them respectively.

Table 4 Recruited subject information

Criteria	Number of subjects	
Dominant hand	Right	21
	Left	3
Watch wrist	Right	16
	Left	8
Sex	Female	13
	Male	11
Age group	18-20	5
	20-24	9
	25-29	4
	30-34	2
	35-39	1
	45-49	2
	55-60	1

The proposal of human subject recruitment and data collection processes was submitted to, and approved by University of San Francisco, Institutional Review Board (IRB) for the Protection of Human Subjects.

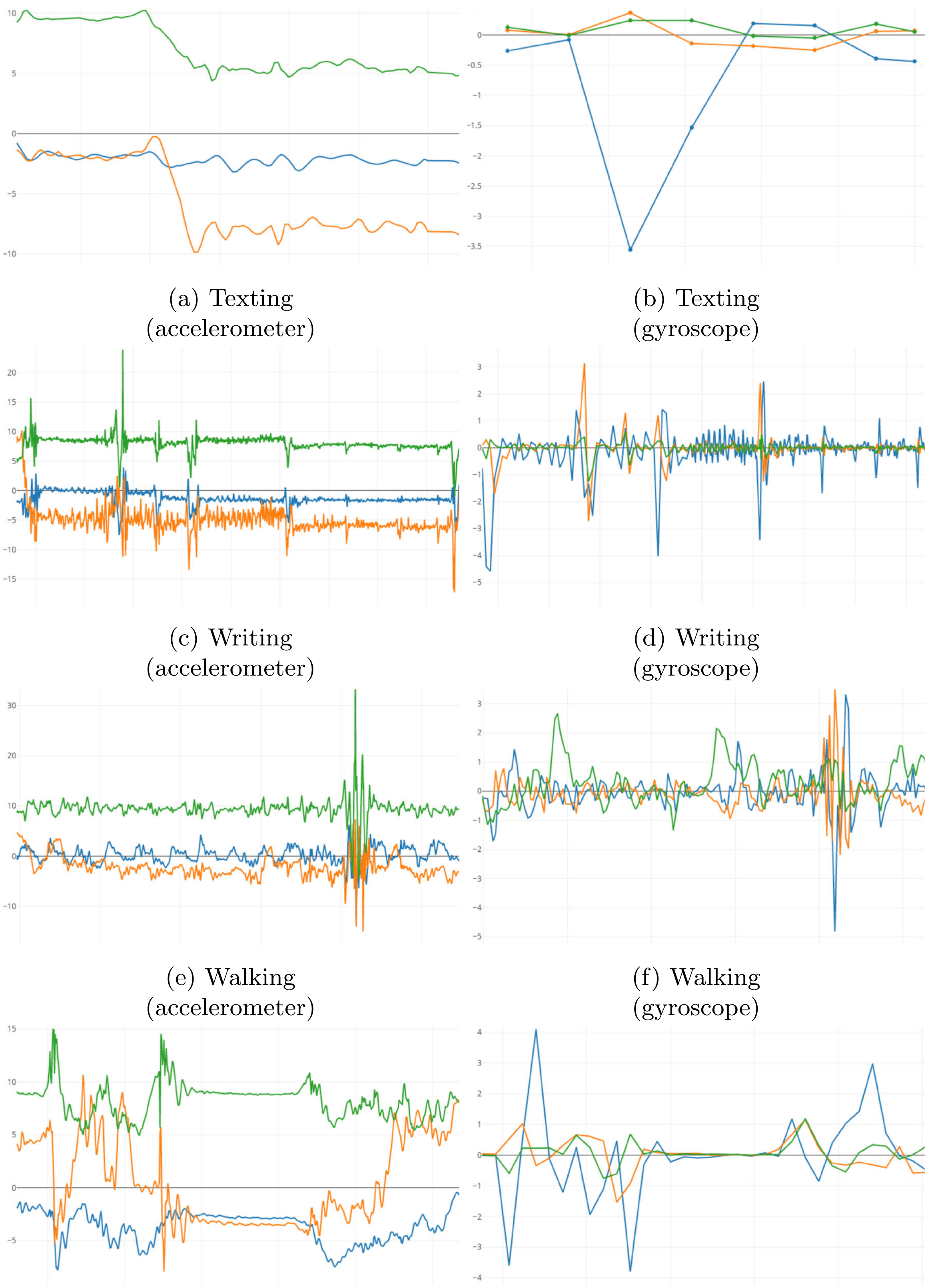
Example accelerometer and gyroscope readings during medication intake and other activities are given in Figs. 3 and 4. In the example given, the subject was wearing the watch on the left wrist which is the subject's non-dominant wrist.

Results

To evaluate the performance of our models, we compared the model fitting time and the F1 score. The F1 score is a measure of prediction accuracy, considering true and false positives and negatives, where 1 is the best and 0 is the worst. For a highly imbalanced dataset, F1 score is a better

Table 5 Activity Types and Watch Wrists (Each subject repeated each activity five times)

Activity	Activity class	Watch wrist
Pill medication intake	Medication intake	Left
		Right
Liquid medication intake		Left
		Right
Texting	Non-medication	Preferred Wrist
Writing	Intake	by the Subject
Walking		
Bottled water intake		



(a) Texting (accelerometer)

(b) Texting (gyroscope)

(c) Writing (accelerometer)

(d) Writing (gyroscope)

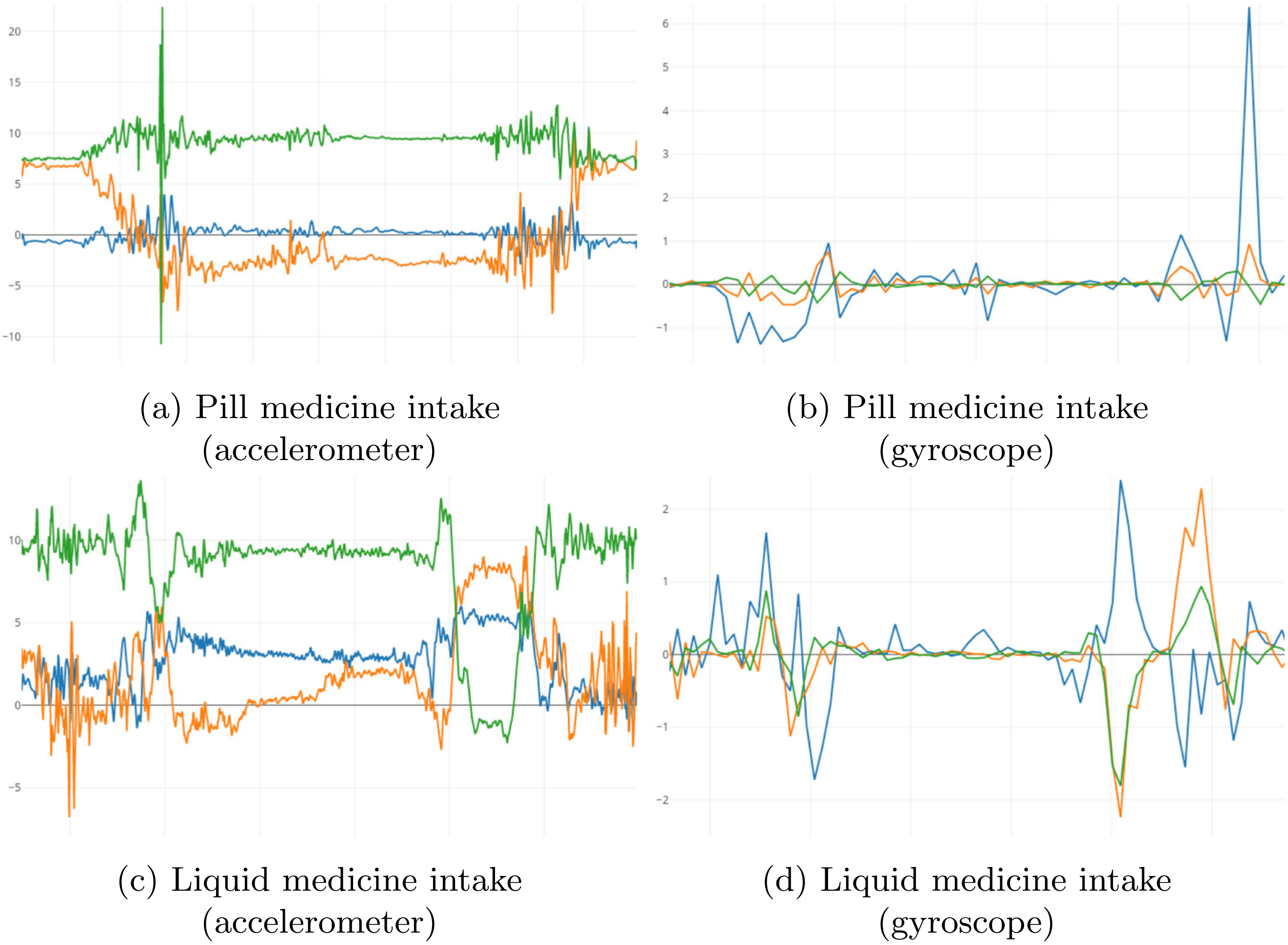
(e) Walking (accelerometer)

(f) Walking (gyroscope)

(g) Bottled water intake (accelerometer)

(h) Bottled water intake (gyroscope)

Fig. 3 Accelerometer and gyroscope readings of non-medication intake activities



(a) Pill medicine intake (accelerometer)

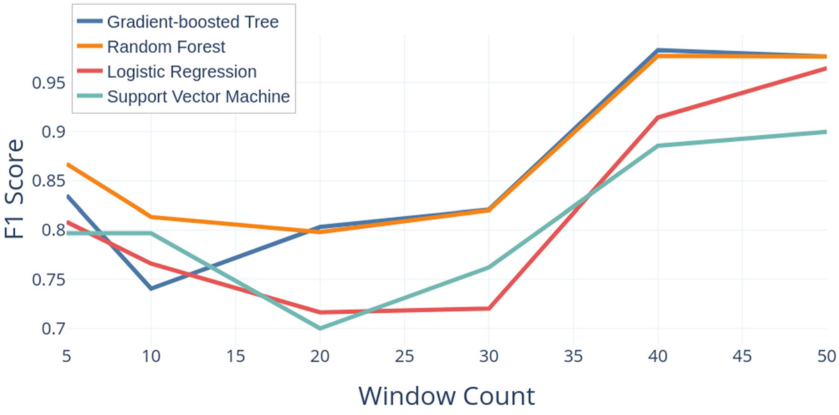
(b) Pill medicine intake (gyroscope)

(c) Liquid medicine intake (accelerometer)

(d) Liquid medicine intake (gyroscope)

Fig. 4 Accelerometer and gyroscope readings of medication intake activities

Fig. 5 Various sliding window sizes and corresponding F1 scores of different machine learning algorithms



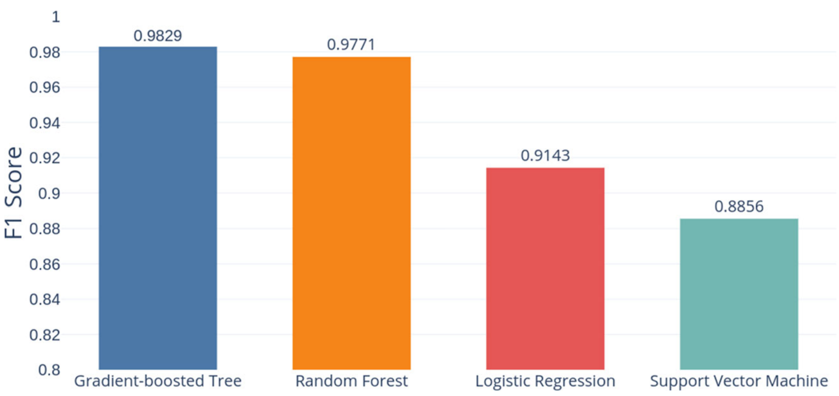


Fig. 6 Different machine learning algorithms and corresponding F1 scores for the window count of 40

measure than accuracy to evaluate a model performance because it accounts for recall and precision.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$
$$Precision = \frac{TP}{TP+FP}$$
$$Recall = \frac{TP}{TP+FN}$$
$$F1 = \frac{2*(Recall*Precision)}{(Recall+Precision)}$$

In order to validate the accuracy of algorithms, we applied aforementioned four different classification algorithms. As the preprocessing step returns different numbers of features depending on the sliding window size, we summarized each data set into 5 to 50 different bins (window count) and calculated F1 scores. Figure 5 shows window count and F1 score of corresponding algorithms and shows that the window count of 40 yields the global maximum for all four algorithms. Figure 6 shows the F1 score of each model where gradient-boosted tree and random forest models yield the highest F1 scores, 0.983 and 0.977, respectively. This results show that the developed system outperforms existing medication intake monitoring systems. Chen's study utilizing inertial sensors with an RGB depth camera achieved an F1 score of 0.9796 using data collected from 5 subjects [12]. Kalantarian's research which required their 25 subjects to wear watches on both wrists achieved an F1 score of 0.4468 due to low precision [30]. Kalantarian's recent study using a smart necklace achieved an F1 score of 0.895 from their 20 subjects [31].

Figures 7 and 8 show the execution time for training and testing each of the machine learning algorithms with a different window count on *Cluster 2*. Although fast prediction time is most critical for providing timely feedback to a user, a medication detection system also requires to train new models quickly. In order to make sure that the developed model is adaptive to a wide range of users with different medication intake behaviors, sensor signatures, and medical conditions, the system needs to re-train a model as more data being collected. In addition, re-training a model will help develop an adaptive adjustment for individuals with changes in medication regimens and medical conditions [54].

Classification models tend to take more time to be trained and tested when the number of windows increase, as this corresponds to the number of features being used. While the gradient-boosted tree model showed the highest F1 score (0.983) when the window count is 40, it takes the longest time (208.784 seconds) to be trained. In contrast, the

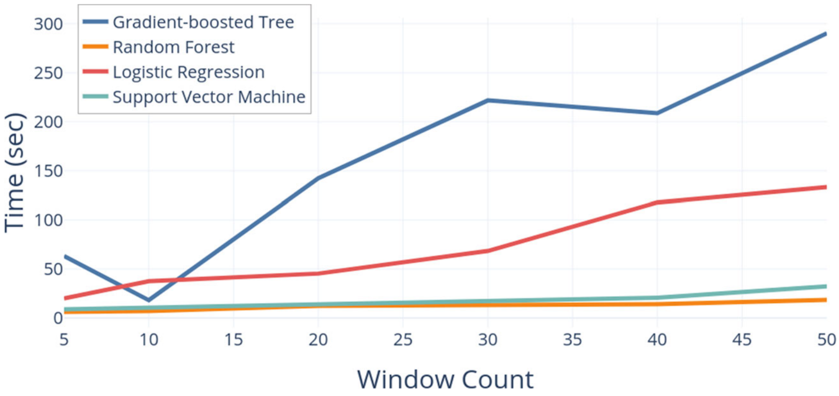
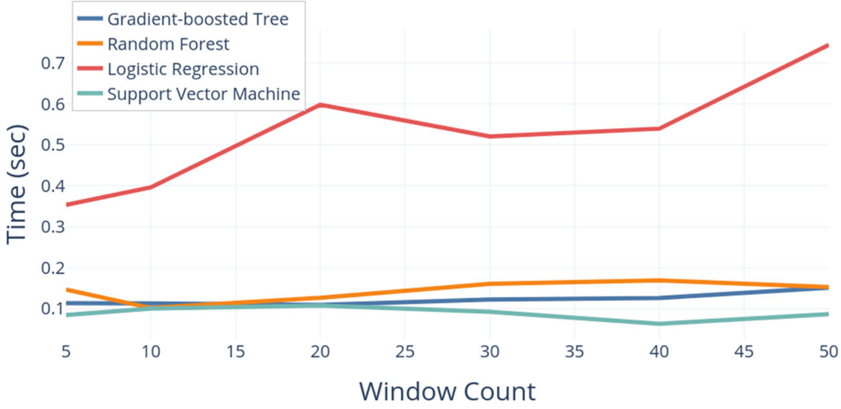


Fig. 7 Various sliding window sizes and corresponding training time of different machine learning algorithms on *Cluster 2*

Fig. 8 Various sliding window sizes and corresponding test time of different machine learning algorithms on *Cluster 2*



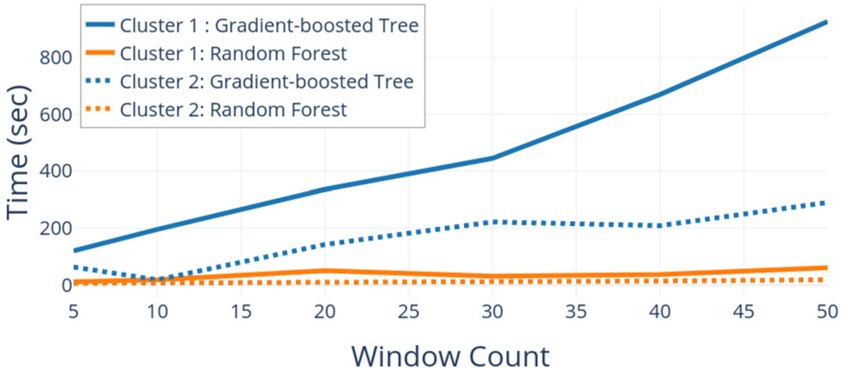
random forest model which has the second highest F1 score (0.977), takes the shortest training time (13.313 seconds).

As *Cluster 1* and *Cluster 2* have different machine specifications including CPU, memory and disk, we compared the training and test time of the two best models, gradient-boosted tree and random forest models. On *Cluster 1*, it takes 36.833 and 0.337 seconds to train and test a random forest model, and 668.909 and 0.482 seconds to train and test a gradient-boosted tree classifier, when the window count is 40. On *Cluster 2*, it takes 14.070 and 0.169 seconds to train and test a random forest model, and 208.784 and 0.126 seconds to train and test a gradient-boosted tree classifier, when the window count is 40 (Fig. 9). Since *Cluster 2* has more computing power including more CPUs, memory and disk space, it showed a better time efficiency. Therefore, the cost of building and training *Cluster 2* is 57.849% of *Cluster 1* and the cost of testing on *Cluster 2* is 48.531% of *Cluster 1*, using random forest and gradient-boosted tree models. When processing data in a distributed manner, data needs to be sent to a number of instances and the processed outcome in each instance needs to be sent back for summarization and this process may require more networking time and overload [27]. Therefore, it is critical to choose and configure a Spark cluster for minimizing time and cost required to build and apply a model. In this case, the data size was large enough that it overcomes the extra networking time and benefits from the distributed and parallelized processing.

Conclusion

In this study, we developed a smartwatch application and cloud-based distributed data storage and processing pipeline for monitoring medication intake. The smartwatch application collects accelerometer and gyroscope data while a subject performs eight different activities and sends the data to a cloud data storage. The developed pipeline processes the sensor datastream and stores the data in a distributed schemaless database, MongoDB. We applied four different classification algorithms to develop distributed machine learning models and compared their F1 scores and training time. The study results show that gradient boosted tree yields the highest F1 score (0.983), although it requires the most training time (208.784 seconds). Alternatively, random forest produced the second highest F1 score (0.977) with the least training time (13.313 seconds). As both gradient boosted tree and random forest algorithms require an insignificant amount of testing time (0.126 and 0.139 seconds respectively), the choice between the two algorithms would depend on priorities between F1 score and training time. The results of our study also show

Fig. 9 Training time of gradient-boosted tree and random forest algorithms using two different clusters listed in Table 3



that a Spark cluster with more CPUs, memory and storage can build a machine learning model faster by utilizing more computing resources concurrently.

Adding extra features using other biosensors embedded in a smartwatch might enhance F1 score, although it would require more training and testing time. In addition to the biosensors utilized in this study, many smartwatches are equipped with NFC which establishes communication and exchanges data between two electronic devices within close proximity (about 10 cm). While our study results show that the applied algorithm could sometimes misclassify data, perhaps applying NFC sensors' data could enhance the outcome. Our future research will also extend the system and clinical study for validating improvements in medication regimen adherence by sending notifications when a subject misses or takes an incorrect amount of medication.

Acknowledgements This work was supported by Jesuit Foundation Grant, University of San Francisco Faculty Development Fund, and Systers Pass-it-on Award by Anita Borg Institute for Women and Technology. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

Funding This work was funded by 1) Spring 2017 Jesuit Foundation Grant, 2) University of San Francisco Faculty Development Fund, and 3) 2018 Systers Pass-it-on Award by Anita Borg Institute for Women and Technology.

Compliance with Ethical Standards

Conflict of interests Diane Woodbridge has received research grants from Jesuit Foundation, University of San Francisco and Anita Borg Institute and has no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with University of San Francisco, Institutional Review Board (IRB) for the Protection of Human Subjects.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

1. Amazon Web Service, Amazon (2019) https://aws.amazon.com.
2. Aldeer, M., Javanmard, M., and Martin, R. P., A review of medication adherence monitoring technologies. *Appl. Syst. Innov.* 1(2):14, 2018.
3. Amazon Web Services, Amazon s3 (2019) https://aws.amazon.com/s3/.
4. Apache Spark, Apache spark: Lightning-fast cluster computing (2019) http://spark.apache.org.
5. Apache Spark, Apache spark: Lightning-fast cluster computing (2019) http://spark.apache.org.
6. Berzati, B., Ippisch, A., and Graffi, K., An android wear os framework for sensor data and network interfaces. In: *2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops)*, pp. 98–104: IEEE, 2018.
7. Breiman, L., Random forests. *Mach. Learn.* 45(1):5–32, 2001.
8. Brown, M. T., Bussell, J., Dutta, S., Davis, K., Strong, S., and Mathew, S., Medication adherence: Truth and consequences. *Am. J. Med. Sci.* 351(4):387–399, 2016.
9. Bruce, P., and Bruce, A, Practical statistics for data scientists: 50 essential concepts. O'Reilly Media Inc., 2017.
10. Chaczko, Z., Mahadevan, V., Aslanzadeh, S., and Mcdermid, C., Availability and load balancing in cloud computing. In: *International Conference on Computer and Software Modeling*, Vol. 14. Singapore, 2011.
11. Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E., Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst. (TOCS)* 26(2):4, 2008.
12. Chen, C., Kehtarnavaz, N., and Jafari, R., A medication adherence monitoring system for pill bottles based on a wearable inertial sensor. In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4983–4986: IEEE, 2014.
13. Chieu, T. C., Mohindra, A., Karve, A. A., and Segal, A., Dynamic scaling of web applications in a virtualized cloud computing environment. In: *2009 IEEE International Conference on e-Business Engineering*, pp. 281–286: IEEE, 2009.
14. Cramer, J. S., The origins and development of the logit model. *Logit Models Econ. Fields* 2003:1–19, 2003.
15. Dean, J., and Ghemawat, S., Mapreduce: Simplified data processing on large clusters. *Commun. ACM* 51(1):107–113, 2008.
16. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W., Dynamo: Amazon's highly available key-value store. In: *ACM SIGOPS Operating Systems Review*, Vol. 41, pp. 205–220: ACM, 2007.
17. Dorman, K., Yahyanejad, M., Nahapetian, A., Suh, M. k., Sarrafzadeh, M., McCarthy, W., and Kaiser, W., Nutrition monitor: A food purchase and consumption monitoring mobile system. In: *International Conference on Mobile Computing, Applications, and Services*, pp. 1–11: Springer, 2009.
18. Furht, B., and Escalante, A., Handbook of Cloud Computing, Vol. 3. Springer, 2010.
19. Gu, L., and Li, H., Memory or time: Performance evaluation for iterative operation on hadoop and spark. In: *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC)*, pp. 721–727: IEEE, 2013.
20. Hansen, R. A., Kim, M. M., Song, L., Tu, W., Wu, J., and Murray, M. D., Adherence: Comparison of methods to assess medication adherence and classify nonadherence. *Ann. Pharmacotherap.* 43(3):413–422, 2009.
21. Hasanuzzaman, F. M., Yang, X., Tian, Y., Liu, Q., and Capezuti, E., Monitoring activity of taking medicine by incorporating rfid and video analysis. *Network Modeling Analysis in Health Informatics and Bioinformatics* 2(2):61–70, 2013.
22. Hayes, T. L., Hunt, J. M., Adami, A., and Kaye, J. A., An electronic pillbox for continuous monitoring of medication adherence. In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6400–6403: IEEE, 2006.
23. Haynes, R. B., McDonald, H. P., and Garg, A. X., Helping patients follow prescribed treatment: Clinical applications. *Jama* 288(22):2880–2883, 2002.
24. Helitzer, D., Heath, D., Maltrud, K., Sullivan, E., and Alverson, D., Assessing or predicting adoption of telehealth using the diffusion of innovations theory: a practical example from a rural program in new mexico. *Telemedicine J. e-health* 9(2):179–187, 2003.

25. Hezarjaribi, N., Fallahzadeh, R., and Ghasemzadeh, H., A machine learning approach for medication adherence monitoring using body-worn sensors. In: *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*, pp. 842–845: EDA Consortium, 2016.
26. Ho, A., Step-by-step android wear application development. Amazon Digital Services, 2015.
27. Howard, A., Lee, T., Mahar, S., Intrevado, P., and Woodbridge, D., Distributed data analytics framework for smart transportation. In: *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 1374–1380: IEEE, 2018.
28. Huang, J. C., Remote health monitoring adoption model based on artificial neural networks. *Expert Syst. Appl.* 37(1):307–314, 2010.
29. Insel, K. C., and Cole, L., Individualizing memory strategies to improve medication adherence. *Appl. Nurs. Res.* 18(4):199–204, 2005.
30. Kalantarian, H., Alshurafa, N., and Sarrafzadeh, M., Detection of gestures associated with medication adherence using smartwatch-based inertial sensors. *IEEE Sensors J.* 16:1054–1061, 2016.
31. Kalantarian, H., Motamed, B., Alshurafa, N., and Sarrafzadeh, M., A wearable sensor system for medication adherence prediction. *Artif. Intell. Med.* 69:43–52, 2016.
32. Laptev, I., Caputo, B. et al., Recognizing human actions: A local svm approach. In: *null*, pp. 32–36: IEEE, 2004.
33. Ma, J., Ovalle, A., and Woodbridge, D. M. k., Medhere: A smartwatch-based medication adherence monitoring system using machine learning and distributed computing. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4945–4948: IEEE, 2018.
34. Mason, L., Baxter, J., Bartlett, P. L., and Frean, M. R., Boosting algorithms as gradient descent. In: *Advances in Neural Information Processing Systems*, pp. 512–518, 2000.
35. MongoDB, Mongodb for giant ideas (2019) https://www.mongodb.com/.
36. Morisky, D. E., Predictive validity of a medication adherence measure for hypertension control, 2008.
37. Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W., Applied Linear Statistical Models, Vol. 4. Irwin Chicago, 1996.
38. Osterberg, L., and Blaschke, T., Adherence to medication. *New England J. Med.* 353(5):487–497, 2005.
39. Pop-Eleches, C., Thirumurthy, H., Habyarimana, J. P., Zivin, J. G., Goldstein, M. P., De Walque, D., Mackeen, L., Haberer, J., Kimaiyo, S., Sidle, J., and et al, Mobile phone technologies improve adherence to antiretroviral treatment in a resource-limited setting: a randomized controlled trial of text message reminders. *AIDS (London England)* 25(6):825, 2011.
40. ur Rehman, M. H., Liew, C. S., Wah, T. Y., and Khan, M. K., Towards next-generation heterogeneous mobile data stream mining applications: Opportunities, challenges, and future research directions. *J. Netw. Comput. Appl.* 79:1–24, 2017.
41. Safavian, S. R., and Landgrebe, D., A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* 21(3):660–674, 1991.
42. Salzman, C., Medication compliance in the elderly The Journal of clinical psychiatry, 1995.
43. Sansone, R. A., and Sansone, L. A., Antidepressant adherence: Are patients taking their medications? *Innov. Clin. Neurosci.* 9(5–6):41, 2012.
44. Seto, E., Cost comparison between telemonitoring and usual care of heart failure: A systematic review. *Telemedicine and e-Health* 14(7):679–686, 2008.
45. Shea, S., Weinstock, R. S., Starren, J., Teresi, J., Palmas, W., Field, L., Morin, P., Goland, R., Izquierdo, R. E., Wolff, L. T., and et al, A randomized trial comparing telemedicine case management with usual care in older, ethnically diverse, medically underserved patients with diabetes mellitus. *J. Am. Med. Inform. Assoc.* 13(1):40–51, 2006.
46. Sokol, M. C., McGuigan, K. A., Verbrugge, R. R., and Epstein, R. S., Impact of medication adherence on hospitalization risk and healthcare cost. *Medical Care*, 521–530, 2005.
47. Speier, C., and Frese, M., Generalized self efficacy as a mediator and moderator between control and complexity at work and personal initiative: A longitudinal field study in east germany. *Human Perform.* 10(2):171–192, 1997.
48. Suh, M. k., Chen, C. A., Woodbridge, J., Tu, M. K., Kim, J. I., Nahapetian, A., Evangelista, L. S., and Sarrafzadeh, M., A remote patient monitoring system for congestive heart failure. *J. Med. Syst.* 35(5):1165–1179, 2011.
49. Suh, M. k., Evangelista, L. S., Chen, C. A., Han, K., Kang, J., Tu, M. K., Chen, V., Nahapetian, A., and Sarrafzadeh, M., An automated vital sign monitoring system for congestive heart failure patients. In: *Proceedings of the 1st ACM International Health Informatics Symposium*, pp. 108–117: ACM, 2010.
50. Suh, M. k., Moin, T., Woodbridge, J., Lan, M., Ghasemzadeh, H., Bui, A., Ahmadi, S., and Sarrafzadeh, M., Dynamic self-adaptive remote health monitoring system for diabetics. In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2223–2226: IEEE, 2012.
51. Suykens, J. A., and Vandewalle, J., Least squares support vector machine classifiers. *Neur. Process. Lett.* 9(3):293–300, 1999.
52. Tucker, C. S., Behoora, I., Nembhard, H. B., Lewis, M., Sterling, N. W., and Huang, X., Machine learning classification of medication adherence in patients with movement disorders using non-wearable sensors. *Comput. Biol. Med.* 66:120–134, 2015.
53. Vlasnik, J. J., Aliotta, S. L., and DeLor, B., Medication adherence: Factors influencing compliance with prescribed medication plans. *Case Manager* 16(2):47–51, 2005.
54. Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., and Petitjean, F., Characterizing concept drift. *Data Min. Knowl. Disc.* 30(4):964–994, 2016.
55. Wu, J. H., Wang, S. C., and Lin, L. M., Mobile computing acceptance factors in the healthcare industry: A structural equation model. *Int. J. Med. Inform.* 76(1):66–77, 2007.
56. Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I., Spark: Cluster computing with working sets. *HotCloud* 10(10-10):95, 2010.
57. Zissis, D., and Lekkas, D., Addressing cloud computing security issues. *Fut. Gen. Comput. Syst.* 28(3):583–592, 2012.

Affiliations

Donya Fozoonmayeh[1] · **Hai Vu Le**[1] · **Ekaterina Wittfoth**[1] · **Chong Geng**[1] · **Natalie Ha**[1] · **Jingjue Wang**[1] · **Maria Vasilenko**[1] · **Yewon Ahn**[2] · **Diane Myung-kyung Woodbridge**[1]

Donya Fozoonmayeh
dfozoonmayeh@usfca.edu

Hai Vu Le
hle10@usfca.edu

Ekaterina Wittfoth
ewittfoth@usfca.edu

Chong Geng
cgeng2@usfca.edu

Natalie Ha
nha2@usfca.edu

Jingjue Wang
jwang193@usfca.edu

Maria Vasilenko
mvasilenko@usfca.edu

Yewon Ahn
yea001@ucsd.edu

[1] Data Science, University of San Francisco, San Francisco, CA, USA

[2] University of California, San Diego, CA, USA