An Intelligent Information Forwarder for Healthcare Big Data Systems with Distributed Wearable Sensors

Ping Jiang[1], Jonathan Winkley, Can Zhao, Robert Munnoch, Geyong Min, Laurence T. Yang

Abstract— An increasing number of the elderly population wish to live an independent lifestyle, rather than rely on intrusive care programmes. A big data solution is presented using wearable sensors capable of carrying out continuous monitoring of the elderly, alerting the relevant caregivers when necessary and forwarding pertinent information to a big data system for analysis. A challenge for such a solution is the development of context-awareness through the multidimensional, dynamic and nonlinear sensor readings that have a weak correlation with observable human behaviours and health conditions. To address this challenge, a wearable sensor system with an intelligent data forwarder is discussed in this paper. The forwarder adopts a Hidden Markov Model for human behaviour recognition. Locality sensitive hashing is proposed as an efficient mechanism to learn sensor patterns. A prototype solution is implemented to monitor health conditions of dispersed users. It is shown that the intelligent forwarders can provide the remote sensors with context-awareness. They transmit only important information to the big data server for analytics when certain behaviours happen and avoid overwhelming communication and data storage. The system functions unobtrusively, whilst giving the users peace of mind in the knowledge that their safety is being monitored and analysed.

***Index Terms*—Ambient Assisted Living, Behaviour Monitoring, Hidden Markov Model, Locality Sensitive Hashing, Wearable Sensors, Big Data.**

I. Introduction

The number of elderly and infirm living in sheltered accommodation is increasing, with more people of retirement age in the UK choosing to "age in place" with some form of support - 473,000 in 2008/2009 [1]. On the other hand, in figures calculated by Help the Aged, the number of those actually being supported has decreased by a dramatic 13% in the years 2000 to 2006 [2] with the trend declared likely to continue in successive years. At the same time, AgeUK [2] noted that "17% of older people have less than weekly contact with family, friends and neighbours". These facts and figures show that there is increased risk for those not being monitored or personally cared for: from minor incidents in the home, from illness which causes immobility or from other unforeseeable scenarios which as such would go undetected if no contact is made with the individual over a long period of time.

For a considerable time, many assistive devices have been available for installation into residential environments or for wearable sensors with the intention of interacting with a user to ascertain their wellbeing, or in some cases their physical health [3, 4]. Elderly monitoring systems can be categorised to two variations: autonomous problem-determining and human problem-determining. Whilst the former category is populated with devices such as Zhou *et al.* [5] and Avci and Passerini [6], these require only the gathered data to infer a belief regarding the users' state. The latter category has the need for an element of further human involvement in order to assess the status of a user. Such applications similarly utilise environmentally located sensors or body-worn nodes [7, 8] to gather readings relating to the user, before uploading them to some ''server'' which is accessible by a healthcare professional or some other monitoring service that can identify any issues being faced by the user. These systems have a lower level of processing involved and as such require heavier data throughput to the server and time-consuming interpretation by healthcare professionals, given that storage of the observations in their raw form is usually required and inference of a behaviour or state is made by a human supervisor. When such healthcare devices need to be deployed to a great amount of the elderly population for continuous monitoring, acquiring and analysing data from the distributed devices become a challenge to data communication and processing. The data generated by the healthcare devices are often semi-structured or unstructured and have the 3Vs characteristics of big data, i.e. Volume, Velocity, and Variety [9]. As a consequence, much of the value of the data is not currently being fully appreciated and used in the healthcare sectors.

This paper presents a big data pilot system for healthcare of the elderly that combines the two categories, i.e. autonomous problem-determining and human problem-determining, and covers the services of both continuous behaviour monitoring and long-term health condition analysis. The system consists of a wrist-wearable sensor node for information collection, a mobile phone for user interaction and remote access, and a centralized big data system as a tool for health condition

[1] Corresponding author: phone +44 1482 465680 Fax +44 1482 466666.

P. Jiang, C. Zhao, and R. Munnoch are with Department of Computer Science at University of Hull, HU6 7RX, UK (e-mail: p.jiang@hull.ac.uk, c.zhao@hull.ac.uk; r.munnoch@2011.hull.ac.uk).

J. Winkley is with the School of Computing, Informatics and Media at University of Bradford, BD7 1DP, UK (e-mail: j.j.winkley@bradford.ac.uk).

G. Min is with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK(e-mail: G.Min@exeter.ac.uk)

L.Yang is with Department of Systems and Computer Engineering, St. Francis Xavier University, Ottawa, ON, Canada (e-mail: ltyang@stfx.ca).

The work was partly sponsored by iMonSys Ltd, UK.

monitoring. For managing such a system, there is a trade-off between distributed processing in the wearable sensors and the centralized analytics in the server cluster. Thus, an intelligent information forwarder embedded in the mobile devices is proposed in this paper to monitor the behaviours of a wearer continuously, alert a caregiver if any anomaly is detected, and transmit only the interesting information to the healthcare big data system for analysis. The intelligent information forwarder based on a Hidden Markov Model (HMM) makes the distributed sensors context-aware and greatly reduces the communication loads and data storage for a large scale system.

With the ability to recover a hidden state sequence from only the visible observations, the HMM is utilised in a broad spectrum of applications. Within the bioscience field, for example, the model is ideal for gene prediction - where each state emits random DNA strings of random length, which are observable as a means to determine the gene producing them [10] - and in protein structure prediction and genetic mapping [11]. Cryptanalysis and cryptography benefit significantly from the utilisation of the HMM [12]; and in the measurement of partial discharge (PD), the time-varying and sequential properties lend themselves to be modelled with an HMM such that PD patterns can be classified to inform of insulation system defects [13].

The traditional HMM uses probability distributions or discrete probability values assigned to single observations. In the behaviour recognition task, more detailed models take observations from a variety of sources to ascertain an intelligent estimate of the hidden state. When the hidden state can be determined with greater accuracy if a number of observation sources are reviewed, e.g., the wearable sensors developed in this paper, the fusion of such inputs must be considered [14-16]. What must be taken into consideration, however, is that this fusion of multiple sensors can in some cases produce worse results than the output of the best single sensor. This can be due to the possibility of inaccurate sensor readings being combined with those evaluated to be more accurate [15]. Nonlinear and high dimensional issues of the sensor readings [17, 18] also can contribute to this.

This paper proposes a sensor fusion scheme to estimate the observational probability of states for HMM based user behaviour detection utilising the developed wearable sensors. It uses a Locality Sensitive Hashing (LSH) table to carry out Instance Based Learning (IBL). Experiments are conducted to compare the performance of the proposed method with the nonlinear dimension reduction method[18] and the results show that the proposed scheme is more efficient for both learning and querying. It is obvious that such intelligent processing embedded in a mobile device should take a resource saving approach due to the limited memory, computational power and communication bandwidth available on board.

The rest of this paper is organized as follows. The system architecture and software are described in Section II, including the details of operational processes and the signal processing for robust measurement. Section III presents the HMM based state and anomaly identification that is the key component of the intelligent forwarder. Section IV explains how LSH can be used as an efficient mechanism to estimate the user's state from the captured multiple sensor signals using probabilistic modelling. Section V presents the developed prototype system and results obtained from the system, which are compared with another commonly used method, i.e. dimensional reduction method. Finally, Section VI contains the conclusions drawn from application of the device in the test scenario.

II. A BIG DATA SYSTEM FOR HEALTHCARE OF THE ELDERLY

Public healthcare is facing serious difficulties due to the rapidly growing ageing population. These individuals have a desire to live independently rather than relying on intrusive care and support. They are also at a higher risk of suffering from illness, accidents, and injuries in their day-to-day activities. Consequently, there is a need for a system that can be conveniently wearable to monitor vital physiological parameters and check health conditions of a user, whilst communicating with the health service providers. The users are dispersed in the whole country and with enormous diversity.

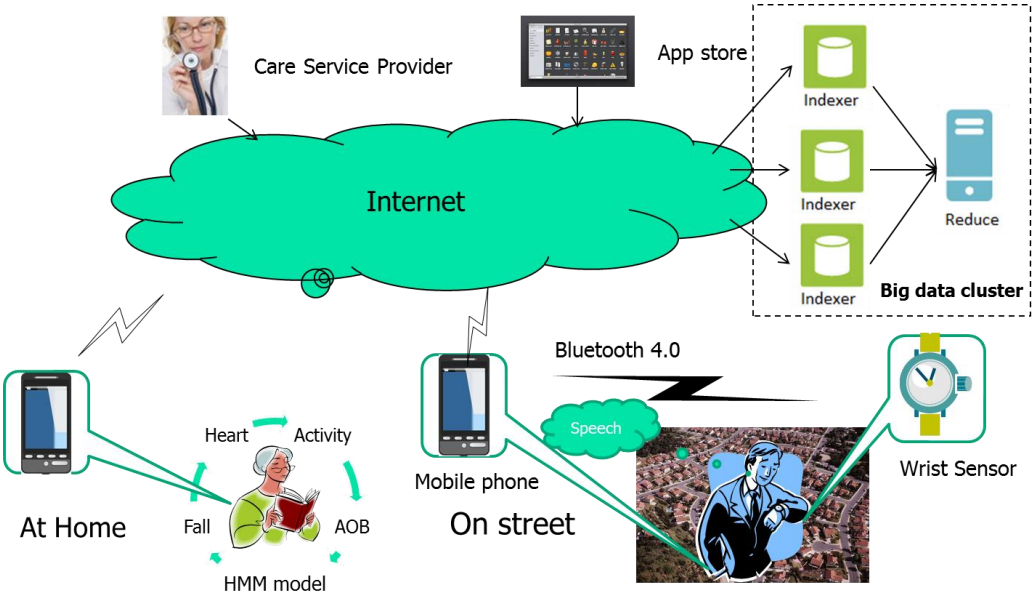

Fig.1. System architecture

Managing such a diverse user group is a challenge faced by the health service providers. The mobile computing and big data infrastructure are opening a new era to next generation healthcare. Individual users can access a tailored and instant health service from the big data system. There can be a great variety of services, for example, daily health check, medication reminder, first-aid instruction, comparative effectiveness research, preventive care, and healthy lifestyle encouragement. Some applications can be downloaded from cloud to a mobile device to provide instant responses to emergency situations. Some others may be computationally intensive in order to analyse a huge amount of sensor data for a long-term healthcare service. Therefore, design of a big data system for healthcare should have a trade-off between distributed intelligence and centralised data analytics.

This paper presents the prototype of a big data system for healthcare of the elderly. It can improve not only the long-term care of this population but also increase the efficiency of healthcare through the integration of distributed monitoring with centralised analytics. The developed system includes three separate components: wrist device, mobile phone, and a big data cluster, as shown in Fig 1. The first version of the system, *Verity*, was reported in [18], which included a customised wrist device and mobile phone but without the centralised big data system. This paper reports the 2nd version of the system for linking wireless measurement with a centralised big data system.

A. Wrist device

The new wrist device has been redesigned to include more sensors and use Bluetooth Low Energy (BLE) technology for connecting with an Android phone to form a PAN (Personal Area Network). It was developed by using TI CC2540, as shown in Fig 2, which is a System-on-Chip (SoC) with BLE support. The wrist device board includes an accelerometer to measure activities of the wearer, a temperature sensor to measure ambient temperature, a thermopile to measure skin temperature, and two reflective photoplethysmography sensors to measure heartbeat and SPO2 in the blood.

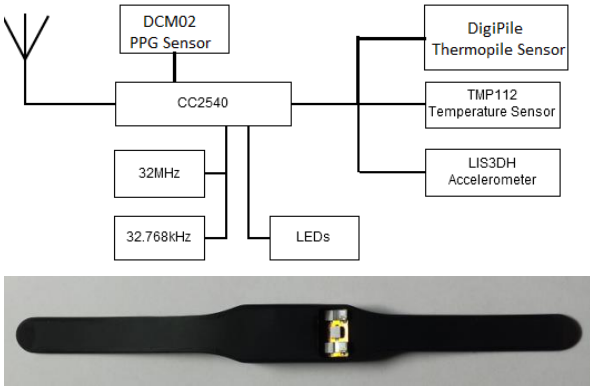

Fig. 2. Schematic diagram and picture of the wrist device

The thermopile, temperature sensor and accelerometer have digital serial interfaces for the CC2540 to read. The PPG (photoplethysmography) sensor is controlled by an analogue switcher to choose the type and intensity of the illumination as red (660nm) or infrared (905nm) for heartbeat rate and SPO2 measurement. An adaptive threshold algorithm was developed for robust measurement of the heartbeat rate.

The adaptive threshold algorithm was an effective extension to the peak detection method proposed in [19], which used a threshold with a decay constant. The PPG signal is a very dynamic signal which can be subject to great variability in the amplitude from cycle to cycle. According to [19] this variability is due at least in part, to the combination of respiratory cycles and motion changes. The adaptation in their method was to allow the decay constant to vary with the sample frequency, the standard deviation of the signal and the amplitude of the previous peak p_{n-1} . The first term is constant for any particular sampled signal, the second term does vary but only slightly given a reasonable time frame to reduce noise; and so effectively the only adaptive term was the previous peak height, with no adaption for the timing of the signal used.

The main idea to improve their algorithm was to extend the adaptive decay constant. The extension was to allow the previous cycle characteristics to predict a height threshold at the next peak arrival time. This sets the decay constant accordingly and adaptively at every cycle. So the new definition of the decay rate D_k is

$$D_k = \frac{(p_{n-1} - Pmin_{n-1})H}{Tbp_{n-1}} \quad (1)$$

$$Pmin_{n-1} = \frac{1}{L}\sum_{i=1}^{L}(p_{B\,n-i}) \quad (2)$$

where p_{n-1} is the last peak greater than the threshold. $Pmin_{n-1}$ is the estimated noise floor that is estimated by the average bad peaks detected. Tbp_{n-1} is the period of the previous heartbeat. H is the coefficient to determine the decay rate. L is the number of bad peaks, p_B, to look back over.

The thresthold is therefore decayed with time t as

$$T(t) = p_{n-1} - D_k t \quad (3)$$

Any detected peaks lower than $T(t)$ are classified as bad peaks that are used to estimate the noise floor in Eq. (2). The first peak greater than the threshold $T(t)$ is classified as the good peak, p_n, for the heartbeat rate calculation:

$$HB(n) = \alpha \times HB(n-1) + (1-\alpha) \times 1/(t(p_n) - t(p_{n-1})) \quad (4)$$

where α, $0 \le \alpha \le 1$, is the coefficient of the first order low-pass filter.

The adaptive threshold is robust to noise because it reduces the decay rate if estimated noise floor P_{min} is high, which means a peak has to overcome a higher noise floor in order to be considered as a valid peak. It is also robust to false peaks due to motion changes between heartbeats because it adjusts the sensitivity of the peak detector by taking previous period Tbp_{n-1} as a reference.

The SPO2 can be calculated and given by the ratio of the two reflected intensities from the PPG sensors [20]:

$$R = \left(\frac{AC_{red}/DC_{red}}{AC_{IR}/DC_{IR}}\right) \quad (5)$$

where AC_{red}, AC_{IR} are the peak to valley amplitude characteristics of the received red and infrared light intensity respectively; DC_{red}, DC_{IR} are the average amplitude of received light under red and infrared respectively. The SPO2 value can be obtained by a lookup table using the R.

B. Mobile app

The wrist device sends the measured parameters to the mobile phone through BLE communication. An app for Android phones was developed to process the gathered data and make a subsequent decision.

The mobile phone houses the data gathering function and main intelligence of the system, as shown in Fig 3. It receives sensor readings from the wrist device (Fig 3.a) with sampling frequencies controlled by different timers, for example acceleration every 0.1 sec, skin temperature and RSSI every 2 sec, heartbeat and SPO2 every 3 sec, and ambient temperature every 10 sec.

The app also enables intelligent behaviour recognition for instant and unobtrusive care as shown in Fig 3.b, which will be discussed in detail in Sections III and IV. It recognises the states of a user and controls voice-based human machine interaction when an anomaly is detected, which is mainly for avoiding false-positive detections. In this scenario, the user is alerted of a situation by communication (through the speaker) from the device, which is preloaded with a series of statements or questions related to a number of scenarios possible during its use, as shown in Fig 4 for a detected fall. The alert follows a decision tree where at each stage the user is required to either confirm or deny a statement, causing the device to adjust its operation accordingly. The states include observable states and hidden states, which are *onTable*, *Fall*, *Nolink*, *Link*, *Abnormal*, *Sleep*, *Sit*, *Stand*, *Walk*, *Run*, *Turn*, *Tap*, *LowBattery*, *Call*, *Text*. The observable states, such as *Fall* and *Nolink* (referring to *no* communication between the wrist device and the mobile phone), can be determined from sensor and component readings directly, with little to no algorithmic processes. The typical result of the majority of the *Fall* and *Nolink* states is to start a voice dialog and dial-out if needed.

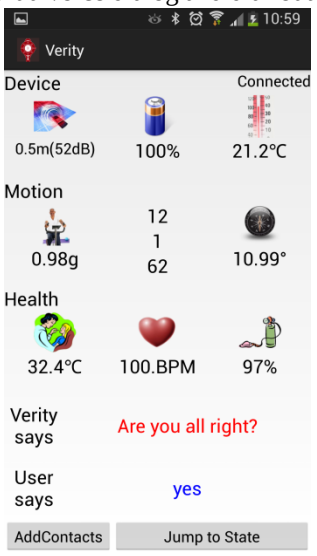

Fig. 3.a Sensor information

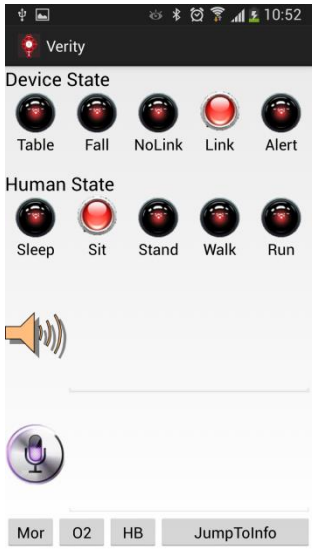

Fig.3.b State recognition

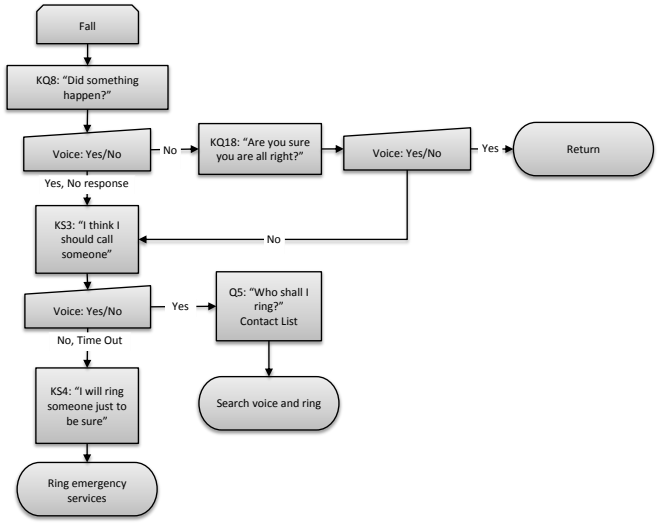

Fig. 4 Speech dialog tree used to identify the necessity for calling for assistance in the event of a detected fall

The hidden states, e.g. *Sleep*, *Sit*, *Stand*, *Walk*, *Run*, and *Abnormal* are estimations of the inferable behaviours of a user which are not explicitly determinable from the sensor readings alone. A behaviour classifier is developed in this paper for their detection.

C. Big data server

The sensor readings and the states of a user need to be sent to the big data system for analytics, which can improve and personalize the quality of care, guarantee efficient use of scarce health professional expertise, and provide statistic evidence for government strategic planning. There is also a potential to reach rural patients without proper access to healthcare and to ensure that patients know when and how medication should be adjusted.

In order to enhance efficiency for large scale and unstructured data retrieval and analysis, A Mapreduction model [21] is used for parallelization with eliminated synchronization problems as shown in Fig.1. MapReduction is a software framework introduced by Google in 2004 to support distributed computing on large data sets using a cluster of computers. It has been widely used as a standard model in big data systems. The big data cluster includes several indexers in parallel and a reduction server for search and statistic operations. It is designed to receive the data stream from mobile phones through a TCP or UDP ports. Usually UDP is not desirable to transmit critical signals because it does not guarantee a delivery. However, in some applications with high velocity of data, UDP can be more appropriate than TCP if additional delivery checking is implemented over it. In addition, the system is flexible enough to input various machine generated data streams in various formats, which can be in log files, CSV files, databases, networking messages, and through scripts. This allows the system to connect with a large number of distributed information sources with non-standard data and unpredictable formats, such as from hospital websites, medical data archives, and diagnostic equipment. Data mining and pattern recognition algorithms can be developed to achieve context awareness from

distributed information for historical behaviour analysis, health condition prediction, and anomaly alerts.

A record in a data stream to log information from a user using a mobile phone is shown in JSON as

```
{ “userName”: “David Carroll”
  “deviceAdress”:[12,42,46,68,34,12],
    {
    “time”:”09:20:11 2013/9/12 UK”,
    “eventType”:[Sit],
    “accValue”:[45,23,99],
    “accL1”:167,
    “accAngle”:1.5,
    “RSSI”:-72.4,
    “verityBattery”:90,
    “phoneBattery”:65,
    “ambientTemp”:23.4,
    “bodyTemp”:35.6,
    “location”:[77.134235,-0.4354365],
    “callType”:[0,null]
    “textType”:[1,”Hi,I am Verity. My friend…”],
    “PPG”:[12,127,0,0…127],
    “HB”:83,
    “SPO2”:97,
    “voiceRecord”:[{“q31”,1},{“q32”,2},{“q33”,
    ”neil”}],
    “interface”:0,
    “bleState”:1
  }
```

To support monitoring of many users, each record has a unique 48-bit IEEE address as the identity of a wearable sensor and a username to identify its wearer. Every record includes a timestamp to define a time series of information. The information stored can include events detected by the intelligent algorithm, readings from sensors, geolocation, voice dialog, machine triggered call and text, and so on. If the whole time series of information is sent to the cluster of servers, for example 1 record (assume 1KB/record) sent to the system every 3 seconds, Table I can be used to estimate the amount of storage required by the big data system similar to [21], where the system is expected to manage 10,000 users with a replication factor of 3 to have data redundancy in the big data system.

TABLE I
STORAGE ESTIMATION OF THE BIG DATA SYSTEM FOR 10,000 USERS

Average daily ingest rate	*288GB*	*Users×logging rate × 3600×24*
Daily raw consumption	*864GB*	*Ingest × replication*
1 year	*315TB*	*Ingest × replication×365*
Node raw storage	*24TB*	*12 × 2 TB SATA II HDD*
MapReduce temp storage	*25%*	*For intermediate MapReduce reserve*
Node-useable raw storage	*18TB*	*Node raw storage-MapReduce reserve*

Big data systems can compress incoming data for their storage and index, for example the compressed rawdata file is approximately 10% of the incoming data and the associated index files range in size from approximately 10% to 110% of the compressed rawadata file in Splunk, which is what we used for our implementation. For 10 years running with a 5% growth per year in users, we need to store 4PB data, which needs $4\times 10^3/18=222$ nodes in the cluster at least. Running such a cluster of servers can be very expensive, which requires significant power, cooling, rack space, network port density, etc.

Avoiding over-sampling is important for any big data system design that needs to deal with the properties of 3Vs, especially the high velocity of data from distributed sensors. It is expected that only valuable information is forwarded to the server and ignores the other irrelevant data. An efficient method is to provide remote sensor nodes with local intelligence that feed data to the big data system when an interesting event happens. In this paper, we use an HMM based hidden state estimation to schedule a data forwarder to achieve context-aware communication.

III. STATE BASED DATA FORWARDER

A big data system manages high volume, high velocity, and/or high variety information assets, which are often from wireless sensors, handhelds, and websites. It is important to develop intelligent data forwarders in individual data sources for feeding meaningful data to the system. This requires a balance between distributed intelligence and centralized analytics in the big data system to avoid missing information or overwhelming the system. Big data systems are often goal/objective-driven. For example, a big data healthcare system can be designed to collect vital parameters of the elderly for understanding general health conditions and exercise engagement through temporal and geographical statistics. Therefore, distributed data sources could be provided with intelligence to determine when and what to feed to the system according to the objectives. This paper develops a data forwarder that is embedded in each data source with context-aware capability, as shown in Fig 5.

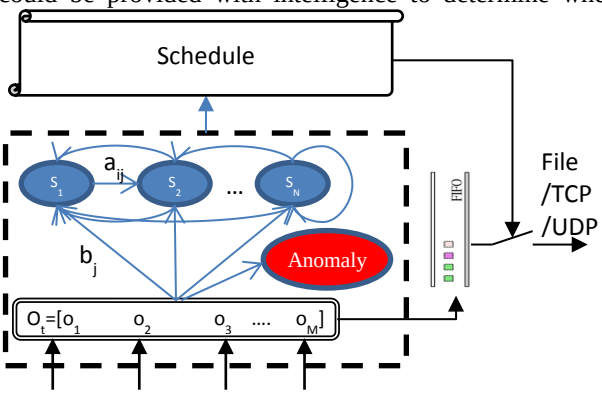

Fig. 5. State driven information forwarder

In this intelligent forwarder, a configurable schedule is developed. The schedule includes a set of rules about the conditions for triggering a voice tree, as discussed in Section II, and logging data to the big data system. According to different analytic objectives, users can specify the rules using meaningful states, for example, “sending sensor data when *running* OR *anomaly* detected OR any *state transition*”. The context-awareness of the forwarder is achieved by an HMM that is used to detect a user’s hidden behaviors, such as *running* and *anomaly*, from its sensor readings.

A. Viterbi algorithm for optimal state estimation

The HMM in Fig 5 has N hidden states, $S = [S_1, S_2, \cdots, S_N]$, and M observations from sensors $O_t = [o_1, o_2, \cdots, o_M]$, t=1…T, where a_{ij} denotes the transition probability, *i.e.*, $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$, and $b_j(O_t)$ represents the observation probability that particular sensor readings O_t are measured in the state j, $b_j(O_t)=P(O_t| q_t=S_j)$.

Given an observation sequence $O=[O_1, O_2, \ldots, O_T]$ and a model $\lambda=(a_{ij}, b_j, \pi_j)$, where i,j=1…N and π_j is the initial probability of state j, the probability of the optimal state sequence $Q^* = q_1^* q_2^* \ldots q_T^*$ can be obtained by Viterbi algorithm [22].

Define

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1, q_2, \ldots, q_{t-1}, q_t = S_i, O_1 O_2 \ldots O_t | \lambda]$$

where $\delta_t(i)$ is the highest probability along a single state sequence as calculated at time t, accounting for the first t observations and terminating with state S_i. The state sequence itself is given in array ψ, which is populated with the state maximizing that probability calculated by δ_t at each step.

(1) Initialize:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$$

$$\psi_1(i) = 0, \quad 1 \le i \le N$$

(2) Recursion Step:

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t),$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right],$$

$$2 \le t \le T, 1 \le i \le N$$

(3) Terminate:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)]$$

$$q_T^* = \operatorname*{argmax}_{1 \le i \le N} [\delta_T(i)]$$

(4) The backtracking procedure:

$$q_t^* = \psi_{t+1}(q_{t+1}^*),$$
$$t = T-1, T-2, \ldots, 1$$

The resulting state sequence, ψ, is the most possible sequence that has emitted the observation at time T, given transitions from previous states.

B. Anomaly detection

The HMM can provide the most likely state sequence based on observations. The probability returned for any state not only provides information about the certainty of the activation of the state, but it can also be interpreted as a value which classifies its degree of anomalousness, where low probabilities denote deviations from the norm [23, 24]. For an identified anomaly, reactions are often to send the current sensor readings to the big data system or contact a caregiver, which can be specified by the user in the schedule. The following 3 types of anomalies are defined in this paper:

*Type*_1 *Anomaly*

A type_1 anomaly is based on the certainty of the winning state. If the probability of the winning state occurring P^* is close to the other states' probabilities, it has very little dominating likelihood of occurring in the current winning state. The proximity to the mean of the probability over all states is calculated as a reference. When the winning probability is close to the mean, the instance can be deemed uncertain.

$$\rho = |P^* - \mu| \le \beta_1, \text{ where } \mu = \frac{1}{N} \sum_{i=1}^{N} \delta_T(i) \qquad (6)$$

In the case where the value of ρ falls within a specified threshold β_1, it indicates significant uncertainty of the identified state. The illegible state means a wrongly defined model that faces an unmodeled state or needs model parameter re-estimation using the Baum-Welch algorithm [25].

Type_2 Anomaly

An equally likely scenario develops when the observation witnessed does not belong at all in the sequence. Detecting such an error primarily requires monitoring of the relevant observation probability. If the probabilities over all states having seen observation O_t is low, the inference is that the model has not seen such an observation before and therefore requires either reassessing or triggering an alert.

$$\sum_{j=1}^{N} b_j(O_t) \le \beta_2 \qquad (7)$$

An instance where this form of anomaly could occur is likely, if not all of the possible observations and associated states were captured during the training phase, or if the user exhibits a behavior typical of an unprogrammed state which is subsequently required to be included. In an instance where the observation is indicative of a serious issue with the user, *e.g.*, a stroke or heart attack indicated by an increase in temperatures and heart rates, the observation would trigger this type of anomaly due to the state not having been seen during training.

Type_3 Anomaly

A type_3 anomaly is a slight variant on the type_2 anomaly and can occur simply when the state at a time step differs for each state determining method within the HMM e.g. the Viterbi state q_t^* and the winning state according to pure observation probability $b_j(O_t)$ do not match significantly. For example, if the observation probability is highest for perhaps the state of *Running*, yet the determined state according to the Viterbi method q_t^* returns *Sleeping* with much higher probability over its *Running* probability, this may in fact indicate a period of distress for the user such as in the instance of a heart attack or some other such observable problem. The probability from Viterbi is first normalized as

$$\hat{\delta}_t(j) = \frac{\delta_t(j)}{\sum_{i=1}^{N} \delta_t(i)}, \; j = 1 \ldots N \qquad (8)$$

If $q_t^* \ne \operatorname*{argmax}_{1 \le j \le N} \left[b_j(O_t)\right] \equiv q_t^O$, a type_3 anomaly is identified by:

$$|\hat{\delta}_t(q_t^*) - \hat{\delta}_t(q_t^O)| \ge \beta_3 \qquad (9)$$

where β_3 is a threshold to identify whether or not the difference between the two differing states is significant enough to trigger an alarm.

As well as identifying possible occurrences of serious health problems, when viewing the entire state determining process as a whole sequence - perhaps after a significant period of monitoring - this type_3 anomaly will prove quite useful for the

detection of behavior changes as it has the potential to highlight instances where the user exhibits a behavior not considered likely according to the transition probabilities programmed at the start of the process. Where a non-threatening state is observed (*i.e.*, the user has in fact begun a higher level of exertion immediately from a rest period, thereby triggering a *Sleeping* to a *Running* state change) then the transition probability between the two requires amending to allow for such an observation sequence.

The schedule in Fig 5 can be configured to select under which states or anomalies the sensor data should be sent to the big data system for analytics. In order to avoid missing important information when an event happens, an FIFO buffer is used to hold a series of latest information and will be sent to the big data system once fired by the schedule.

The context-awareness of the intelligent forwarder relies on correct behaviour detection. In case of an outdated Markov model, detected states could be wrong and important information could be missed. It will cause an increasing number of abnormal behaviours to be detected, which may be due to health problems or due to outdated models. Thanks to the voice verification mechanism of the system, false anomalies can be easily identified and used to trigger a modelling process for learning HMM parameters, such as using the Baum-Welch algorithm.

IV. Instance Based Learning of Observation Probability

The HMM in Section III defines two probabilities, i.e. transition probabilities a_{ij} and observation probability $b_j(O_t)$ representing the probability that state j has observation O_t. Utilising these probabilities it is possible to identify the most probable state at a specific time step based on the observations made at that point along with the preceding states. It is also able to provide a solid estimate of the most likely state sequence for an entire set of observations over a prolonged period of time. As the observation O_t includes readings from multiple sensors, determining the observation probability becomes more difficult due to involving high dimensional similarity measures. In terms of the wrist device developed in this paper, the dimensionality of the sensor readings can reach eight, which includes skin and ambient temperatures, heartbeat, two PPGs, and accelerations in three axes. There is also a considerable chance of nonlinearity between data clusters present because the data may lie on nonlinear manifolds, which make classification based on data distance unreliable given its tendency to misrepresent true topology. Physiological parameters often have such inherent nonlinearity, for example, acceleration and heart rate exhibits a hysteresis relation.

The greater the number of data attributes (dimensions) the lesser the ability to make sense of the data due to the fact that with nonlinearity in a higher dimension, standard Euclidean distance functions lose their usefulness and so clustering with such methods becomes less accurate.There are a multitude of techniques for dealing with nonlinear, high dimensional data, with many sharing basic underlying principles to reach the lower dimensional representation of a complex nonlinear data set: Sammon's mapping[26], Isomap[27], Curvilinear Component (and Distance) Analysis[28, 29] all seek to replicate similar distances between points located in a high dimension after placement in the lower dimension, by a means of gradient descent or iterative error reduction methods.

A Curvilinear Distance Analysis algorithm was presented by the authors in [18] for determining the observation probability $b_j(O_t)$. The observation O_t may be in a high dimensional and nonlinear space. If it lies on a nonlinear manifold, Euclidean distance makes less sense for classification but has to be replaced by Curvilinear Distance to measure the distance along the manifold. The algorithm unfolds high-dimensional manifold data to low dimension by retaining topology and forces the clusters to be linearly separable. The algorithm's effectiveness was validated by experiments using the *Verity* platform, however, it is quite time consuming for the data unfolding because it involves intensive computation to project prototypes in high dimension space to a low dimension space and maintain equivalent curvilinear distances. Sometimes, such equivalence may even not exist. Instance Based Learning is proposed in this paper as an alternative to facilitate learning of $b_j(O_t)$ from demonstration.

A. Locality sensitive hashing for instance based learning

Instance based learning (IBL)[30] takes directly sampled data from any system at a known state and constructs a hypothesis regarding similarity without the need to generalise a model based on the often high dimensional and nonlinear data. Through learning, data instances are stored in some form of memory. This is then accessible for subsequent classification operations, where a query is submitted and compared with all trained values according to some distance metric in order to ascertain its membership to the encoded classes. IBL has multiple advantages over parametric and model-based algorithms, especially in the storage of new, unseen instances. Other algorithms would typically require a complete re-examination of the data set in order to be wholly inclusive of the new data points where IBL methods simply "insert" the new data instance without disrupting any previously determined model.

It is commonly accepted that the genus of and starting point for IBL algorithms are the simple *k*NN(*k*-Nearest Neighbour) classifier[30]: saving training instances to some data structure such that other instances may be compared distance-wise with those local data already classified to return a possible containing state for the new instance [31]. As highlighted in [32], for large data sets with high dimensionality (M), searching through n instances of a data set in order to determine those within the closest proximity can take an extensive amount of time, given that all pairs require evaluation using a distance measure such as Euclidean or Hamming.

Locality Sensitive Hashing (LSH) [33, 34] provides adequate means to speed up the process of nearest neighbour searching, overcoming the above issue by storing the data in another variable-tolerance, compressed format which is easily searchable and requires only simple look-up operations to determine possible immediate neighbours, which can take O(1)

by using E^{2}LSH[34] for example. The principle behind LSH is to hash the sample data in such a way that the probability of two points, p and q, hashing to the same bucket is higher for objects that are close to each other than for those that are further apart.

$$P_H[h(p) = h(q)] \geq P_1 \; for \; \|p - q\| \leq R_1 \quad (10)$$

$$P_H[h(p) = h(q)] \leq P_2 \; for \; \|p - q\| \geq cR_1 = R_2 \quad (11)$$

where $R_2 > R_1 \; and \; P_1 > P_2$.

A family of LSH functions can be defined by p-stable distributions[35], for example projection to linear bins:

$$LSH_h = \left\lfloor \frac{\vec{z}_h \cdot \vec{v} + b}{\omega} \right\rfloor \quad (12)$$

where $\vec{v}$ is the M-dimensional vector to be hashed and $\vec{z}_h$ is a random vector from a p-stable distribution, such as from a $N(0,1)$ Gaussian distribution. Another random value b uniformly in the range $[0, \omega)$ is then added to the scalar projection which is then quantised by ω. ω is the width of the bin in which a data point may fall into. $\lfloor \cdot \rfloor$ is the floor operator.

This paper presents an LSH based IBL for obtaining the observation probability $b_j(O_t)$ from high-dimensional and nonlinear sensor readings. It includes two stages, i.e. learning and querying.

B. Learning

The learning process is to sample typical sensor readings for different states and encode into a hash table H with L independent LSH functions h_1 , h_2, …, h_L defined in Eq. (12). For a given state, S_j, each sampled $O_t(j)$ is first normalised, *i.e.*, $\hat{O}_t = O_t / \|O_t\|$. All sampled sequence of $O_t(j)$, t=1..T, for state j are clustered with a tolerance of ϵ, *i.e.*, if the L_2 distance between any two samples is less than ϵ, they are merged using the k-means algorithm. A set of prototypes $V_t(j)$, t=1,…,T_j , is obtained with $T_j < T$.

The prototypes $V_t(j)$, t=1,…,T_j , are then projected to L bins in Eq. (12) as shown in Fig 6.

Element to be stored

$[X_1, X_2, X_3, \ldots X_M] = V_1$

↓

HASH OUTPUTS

$h_1 = 07$

$h_2 = 09$

↓

	01	02	03	04	05	06	07	08	09	10
h_1	-	-	-	V_9	V_2	V_4 V_7	V_5 V_3 V_1	V_{10}	V_8	V_6
	01	02	03	04	05	06	07	08	09	10
h_2	-	-	V_2	V_7 V_9 V_5	V_3 V_8	V_6	-	V_4	V_1 V_{10}	-

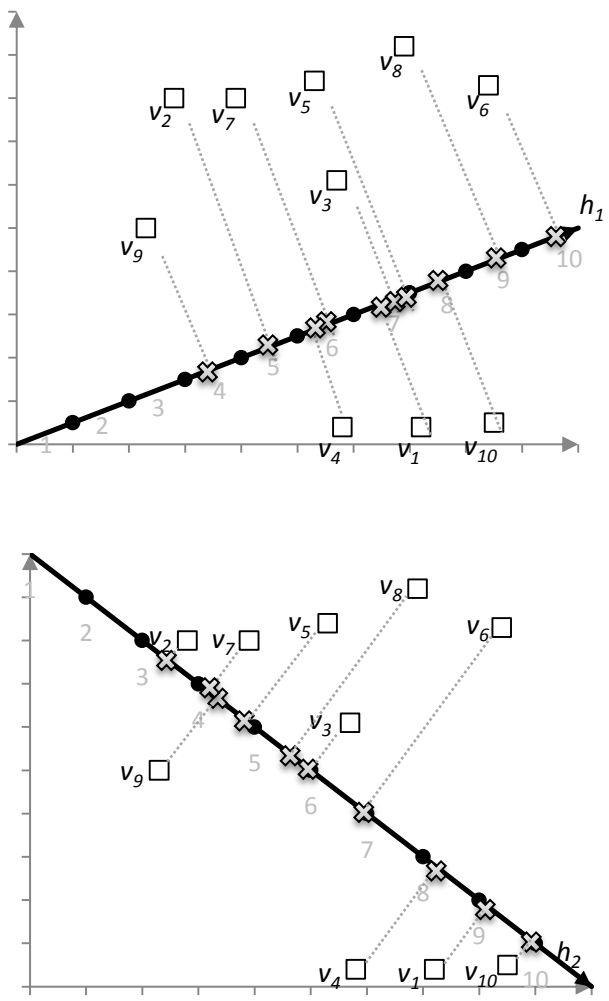

Fig. 6. Construction of LSH table with 10 prototypes and 2 hash functions, h_1 and h_2 where prototypes V_1 … V_{10} are projected into 10 bins(numbered from 1 to 10 in the figure) along two lines h_1 and h_2.

After feeding in $V_t(j)$ for all states j=1, …, N, we have learnt the typical readings of different states. This will be saved for real-time querying about b_j for any sensor reading O_t.

C. Querying

The retrieval method for any O_t is a LSH recall procedure with "bucket" checking. Different from the conventional LSH for *kNN*, we want to calculate the density of observations of a given state j near O_t in a given radius $R \in \boldsymbol{Z}$ for the probability $b_j(O_t)$ estimation.

First, O_t is projected to L bins in the L hash functions in Eq. (12) with $\vec{v} = O_t$, *i.e.*, we have $h_1(O_t)$, $h_2(O_t)$,…, $h_L(O_t)$. The prototypes of state j, $V_t(j)$, encoded in the same bins as O_t are counted $\alpha_1(j)$, where j=1, …, N.

Increasing the searching radius by 1, with additional 2L neighbour bins, $h_1(O_t)\pm1$, $h_2(O_t)\pm1$,…, $h_L(O_t)\pm1$, are checked. The prototypes encoded in them are counted to have $\alpha_2(j)$, where j=1, …, N. The search is expanded to radius R to have the total numbers of prototypes in the radius R, $\alpha_R(j)$, j=1, …, N.

We define a Radius Density of state j with r=1..R:

$$RD_r(j) = \frac{\alpha_r(j) - \alpha_{r-1}(j)}{\sum_{k=1}^{N}(\alpha_r(k) - \alpha_{r-1}(k))}, with \; \alpha_0(j) = 0 \quad (13)$$

The State Weighted Density(SWD) can then be defined by taking into account the distance between the query point and the prototypes, i.e. inverse distance weighting:

$$SWD(j) = \sum_{r=1}^{R} RD_r(j)/r \tag{14}$$

The Observation Probability $b_j(O_t)$ can then be estimated by the state weighted density in the R radius.

$$b_j(O_t) = \frac{SWD(j)}{\sum_{k=1}^{N} SWD(k)} \tag{15}$$

It estimates the likelihood of a state happening by considering the local distributions of the prototypes sampled during the learning stage. This is considered to be robust to nonlinearity and fast in both learning and querying stages, with only hash table insertion and check operations.

V. Testing and Experiments

A. State detection

The LHS based HMM for state identification has been implemented and compared with the dimensional reduction method reported in [18], which can be depicted in Fig 7.

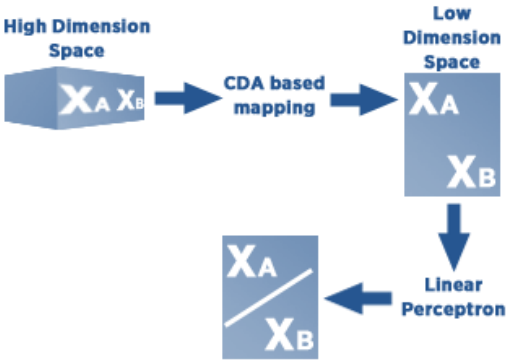

Fig. 7. The dimensional reduction method implemented in [18]

The set of two classes $X = [X_A, X_B]^T \in R^{M\times N}$ is projected to a lower dimension $[X_a, X_b]^T \in R^{M\times n}, n < N$, where the curvilinear distances in a single class are kept, thus "flattening" the high dimension data and linearly separating the clusters X_a and X_b in the lower dimensional space. Based on distance to their closest prototype, successive points can be interpolated efficiently and projected from the high to the low dimension and once separated a simple classifier, e.g. a single layer perceptron, can be used to identify their parent cluster.

The same sensor readings as the experiments in [18] were used for the comparison, which included ambient temperature, skin temperature, heart rate, acceleration magnitude, and its direction with an attributed state which was observed to have produced such readings.

Five states are expected to be identified, which are *S*=[*Sleep, Sit, Stand, Walk, Run*] corresponding to states from 0 to 4. The transition parameters of the HMM remain the same, with a_{ij} specified as in (16) and the initial state probability vector π as in (17), where there is an observed higher likelihood that the starting state is *Standing* over all others.

$$a_{ij} = \begin{bmatrix} 0.45 & 0.35 & 0.20 & 0 & 0 \\ 0.25 & 0.35 & 0.30 & 0.10 & 0 \\ 0 & 0.35 & 0.20 & 0.35 & 0.10 \\ 0 & 0.10 & 0.25 & 0.40 & 0.25 \\ 0 & 0.10 & 0.15 & 0.25 & 0.50 \end{bmatrix} \tag{16}$$

$$\pi = [0.1 \quad 0.2 \quad 0.4 \quad 0.2 \quad 0.1] \tag{17}$$

The readings and the known state that were producing them were first submitted to the dimensional reduction algorithm detailed in [17, 18]. It successfully took the readings from their initial 5 dimensions to the more easily viewable 2, without loss of structure and resulting in the creation of 4 linearly separable state clusters with which subsequent classification of unseen data points can occur (note that the state of *Sleeping* was not observed in this test of *Verity* and data gathering procedure due to the conditions indicating such a state not being easily obtainable during testing). A single layer perceptron network was trained for the classification. Table II illustrates some examples of classification with the perceptron for unseen data points.

TABLE II RESULT OF STATE DETECTION USING DIMENSIONAL REDUCTION METHOD ON UNSEEN DATA POINTS

Ambient	*Contact*	*Pulse*	*Motion*	*Orient.*	*Actual*	*Result (Probability)*
28.699	28.838	80.213	0.000	0	1	1 (0.98)
28.699	28.838	76.142	0.170	0	1	1 (0.98)
28.699	28.849	81.967	0.114	8	2	2 (0.99)
28.699	28.797	81.967	0.458	6	3	3 (0.98)
28.699	28.662	80.213	1.799	0	4	4 (0.98)
28.699	28.704	81.967	1.799	10	4	4 (0.99)

The same training instances were submitted to the LSH table. Table III shows the results, returning 100% classification correctness on the same unseen data as used in the previous experiment.

TABLE III RESULT OF CLASSIFICATION WITH LSH SCHEME FOR UNSEEN DATA POINTS

					State Probability				
Ambient	***Contact***	***Pulse***	***Motion***	***Orient.***	***1***	***2***	***3***	***4***	***Actual***
28.699	28.838	80.213	0.000	0	0.985	0.011	0.003	0.001	1
28.699	28.838	76.142	0.170	0	0.501	0.369	0.000	0.129	1
28.699	28.849	81.967	0.114	8	0.000	0.782	0.198	0.020	2
28.699	28.797	81.967	0.458	6	0.005	0.385	0.498	0.112	3
28.699	28.662	80.213	1.799	0	0.385	0.188	0.035	0.392	4
28.699	28.704	81.967	1.799	10	0.000	0.003	0.014	0.983	4

Table IV details a comparison between the two different state probability determining methods, with key parameters that resulted in the best classification rates during experimentation.

TABLE IV PERFORMANCE TIMES FOR THE STATE DETERMINING METHODS

Method	*Key Parameters*	*Training Time (ms)*	*Classification Time (ms)*
Dimension Reduction with Linear Perceptrons	$tolerable\_loss = 0.1$, $alpha_{\min} = 0.02$, $alpha_{\max} = 0.5$	5713	154
Locality Sensitive Hash Table	R=10, ω=0.001, L=30	32	94

The classification with dimension reduction scheme took 154ms for querying, however it is in the training (projection) of the prototypes that took an outlay of nearly 6 seconds to prototype and project the 30-member training set. Classification is 100% accurate for the experiment, with the returned membership values tending very close to 1 due to the certainty through dimension reduction that the unseen data points fall within the newly created linear boundaries between classes through the perceptron.

The LSH provides a better result over the dimensional reduction methods, with a much shorter training period (32ms) and classification speed (94ms), the 100% correctness and format of probability values seems most appropriate for use in the proceeding Hidden Markov Model as the observation probability. The number of hash functions used in the experiment to produce the results was 30.

B. Healthcare big data system

A prototype of the big data system has been developed by using Splunk Enterprise 6.0 for analytics of the behaviours of wearers, as shown in Fig.8. Splunk is a time-series engine that can collect, index and analyse machine generated data. It can support large-scale data collection and processing with parallelizing analytics via the MapReduce mechanism. Therefore, it can handle distributed information with the 3V characteristics from a great amount of wearable sensors very well.

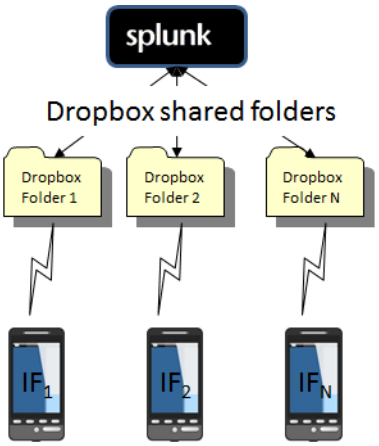

Fig. 8. Architecture of the healthcare big data prototype system

In this prototype system, we used the Dropbox system as a medium to transfer distributed user's information to Splunk engines via WiFi or cellular networks. Each user's mobile phone was deployed with the intelligent forwarder that carries out HMM based state detection continuously as presented in Sections III and IV. The forwarder can be scheduled to log the records or start a voice dialog for alerting a caregiver based on the detected states. Because of the HMM based state detection, the forwarder is aware of the wearer's behaviours and only the records associated with certain events are saved to local files according to the schedule. The files are then synced with a folder in Dropbox by using Android sync API once communication becomes available. If the Dropbox folder is shared with the big data system, Splunk can monitor any changes in the folder and index the data for analytics. It is a concern that big data poses big privacy risks[36]. Therefore, the approach using personal Dropbox folders gives individual users the right to decide if they want to keep the collected information privately or share with someone they trust; for example, they can select to share the folder with caregivers or family members, rather than an insurance company.

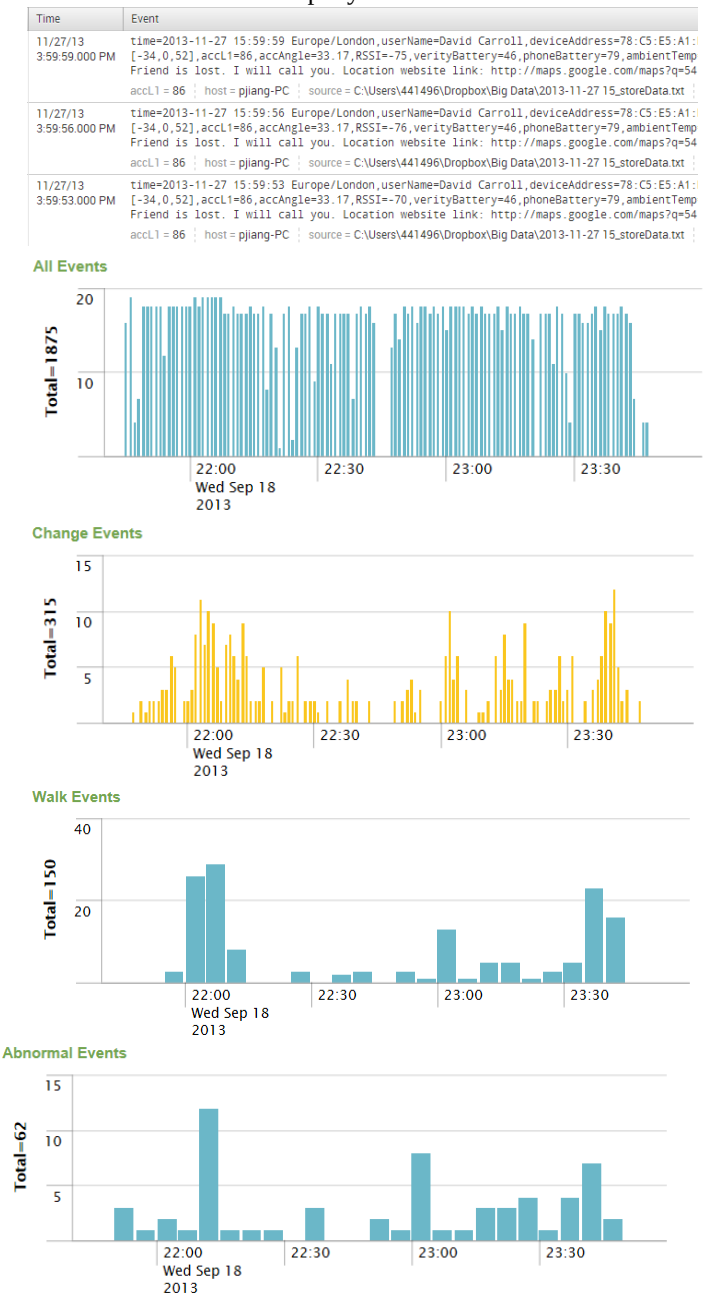

Fig. 9. Forwarding statistics for all events, state changes, walk, and abnormal states

Small scale field trials have been carried out since September 2013 with 3 subjects. An example is shown in Fig.9. A subject, David Carroll, with the wrist sensor was monitored about two hours from 09/18/2013:21:40:00 to 09/18/2013:24:00:00. Without scheduling the forwarder, events were sent to the big data system every 3 seconds, with a total number of 1875 in this period. The forwarder can be scheduled according to the subject's behaviours. If only the information during walking is of interest to a caregiver, 150 records would then be sent to the

big data system. Sometimes the state change could be important; the forwarder can be configured to send only when a change happens, with 315 changes in the example. As discussed in Section III, anomalies can imply an alarm on the health conditions or indicate the HMM is no longer valid which needs a re-estimation of the model. Detected abnormal events should be sent to the big data system for analysis. There were 62 events during the period. A dramatic increasing in anomalies often indicates a poor model to describe behaviours of the wearer and needs a re-calibration. A big data system can be an effective tool to manage distributed models remotely.

As an example, a dashboard with several panels was developed to provide useful clues about a subject's lifestyle and health conditions. Fig 10.a shows the body temperature of the subject, which is an important physiological parameter for healthcare. Fig 10.b illustrates the geolocation distribution of the subject's activities in a month. A change in the distribution usually indicates a change of health conditions, lifestyle or social engagement. Fig 10.c shows the behaviours of the subject during a day. It indicates that the subject did not walk enough as recommended by the caregiver to gain health benefits. A reminder needs to be sent to promote a healthy lifestyle. Fig 10.d shows the average ambient temperature in the home. The system monitors living conditions of the subject that can also provide added value for energy management etc.

The preliminary field trials reported here are only with a small scale and a single server implementation of Splunk Enterprise. However, it is sufficient to prove the concept of the proposed architecture and intelligent forwards to be a big data solution for the healthcare of a great amount of the elderly population. Splunk Enterprise can be deployed into a distributed architecture following the Mapreduction model. It can scale flexibly from a single server to multiple datacentres to cloud, considering the amount of users to be monitored and analysed. Its parallel architecture also means search and indexing performance scales linearly across servers.

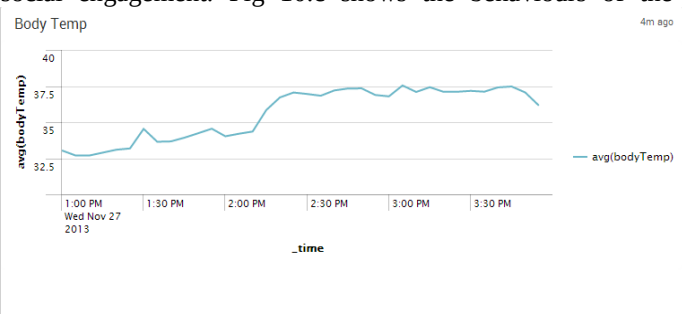

Fig. 10.a Body temperature

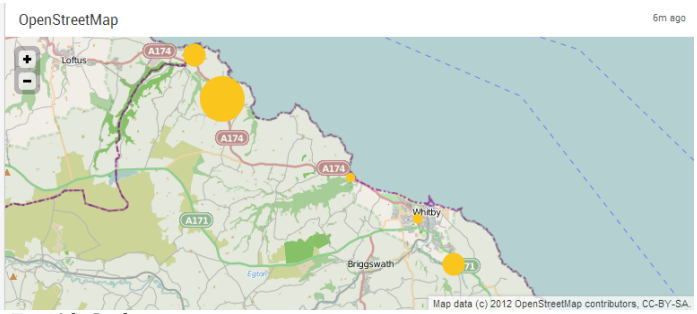

Fig.10.b Geolocation statistics

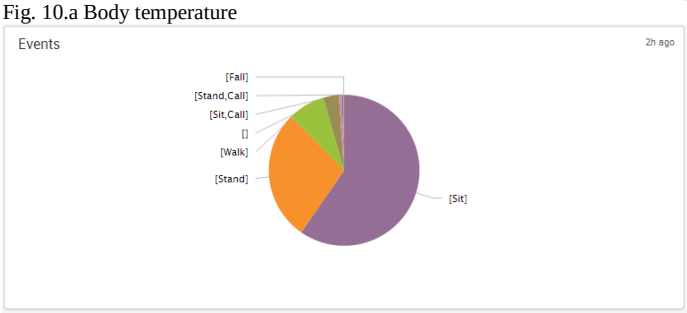

Fig. 10.c State statistics

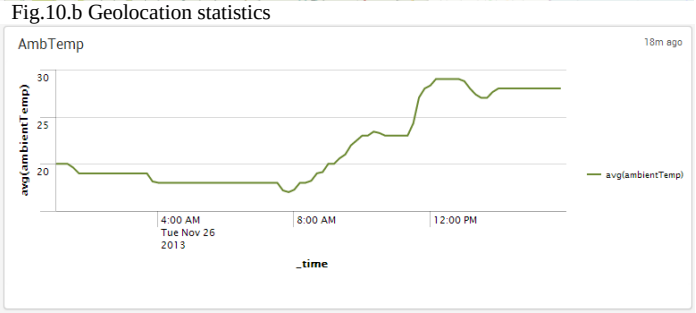

Fig.10.d Ambient temperature

VI. Conclusions

This paper presented a big data healthcare system for elderly people. The system connects with remote wrist sensors through mobile phones for monitoring wearers' well-being. Due to a tremendous number of users involved, collecting real-time sensor information to the centralised servers becomes very expensive and difficult. However, such a big data system can provide rich information to healthcare providers about individuals' health conditions and their living environment. Therefore, this paper proposed an intelligent information forwarder embedded in a mobile phone. It can be configured by a user to determine under which circumstances data should be logged to the system. It uses an HMM to estimate a wearer's behaviours, which includes an LSH table to determine the observation probability of a state. Considering nonlinear and high dimensional aspects of the sensor observations, the LSH table is proposed to improve efficiency. It can be learnt by inserting sample data and queried by checking their local density. Experiments have verified that the LSH based behaviour estimation is more efficient than the dimensional reduction method, which is important for implementation on a mobile device. A prototype of the big data system to work with distributed wearable sensors has been built up for use in the healthcare of the elderly. It demonstrates that the state based forwarder makes the remote sensing context-aware when feeding information to the big data healthcare system.

There could be a large group population of the elderly to be monitored using this system. All of them will have their own behaviour models, e.g. HMMs, about their daily life. Possible future work will be on how the models can be maintained remotely and automatically by the big data system. As section III discussed, frequent false anomalies would be an indication of a mismatching model. With rich information collected in the big data system, the model could be rectified or recreated to fit a user's actual behaviour pattern automatically or through active remote instructions.

REFERENCES

[1] A. S. C. S., "Community Care Statistics 2009-10: Social Services Activity Report, The NHS Information Centre," 2011.

[2] AgeUK Help the Aged, "Older people in the UK," 2008.

[3] H. Yan, H. Huo, Y. Xu, and M. Gidlund, "Wireless sensor network based E-health system: implementation and experimental results," *IEEE Transactions on Consumer Electronics*, vol. 56, pp. 2288-2295, 2010.

[4] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, "Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, pp. 864-873, 2009.

[5] F. Zhou, J. Jiao, S. Chen, and D. Zhang, "A case-driven ambient intelligence system for elderly in-home assistance applications," *IEEE Trans Syst Man Cybern, Part C*, vol. 41, pp. 179–189, 2011.

[6] U. Avci and A. Passerini, "Improving activity recognition by segmental pattern mining," presented at the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Lugano 2012.

[7] L. Ferreira and P. Ambrosio, "Towards an interoperable health-assistive environment: The eHealthCom platform," presented at the 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics, 2012.

[8] V. Venkatesh, V. Vaithyanathan, M. P. Kumar, and P. Raj, "A secure ambient assisted living (AAL) environment: an implementation view," presented at the 2012 International Conference on Computer Communication and Informatics, Coimbatore, 2012.

[9] M. A. Beyer and D. Laney, "The Importance of 'Big Data': A Definition," Gartner, Ed., ed, 2012.

[10] M. Stanke and S. Waack, "Gene prediction with a hidden Markov model and a new intron submodel," vol. 19, pp. 215-225, 2003.

[11] V. D. Fonzo, F. Aluffi-Pentini, and V. Parisi, "Hidden Markov Models in Bioinformatics," *Current Bioinformatics*, vol. 2, pp. 49-61, 2007.

[12] P. J. Green, R. Noad, and N. P. Smart, "Further hidden markov model cryptanalysis," presented at the the 7th international conference on Cryptographic hardware and embedded systems, Edinburgh, UK, 2005.

[13] L. Satish and B. I. Gururaj, "Use of hidden Markov models for partial discharge pattern classification," *IEEE Transactions on Electrical Insulation*, vol. 28, pp. 172-182, 1993.

[14] H. Lee, K. Park, B. Lee, J. Choi, and R. Elmasri, "Issues in data fusion for healthcare monitoring," presented at the The 1st international conference on PErvasive TechnologiesRelated to Assistive Environments, Athens, Greece, 2008.

[15] M. Dong and D. He, "Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," *European Journal of Operational Research*, vol. 178, pp. 858-878, 2007.

[16] L. Atallah, B. Lo, G.-Z. Yang, and F. Siegemund, "Wirelessly Accessible Sensor Populations (WASP) for elderly care monitoring," presented at the Second International Conference on Pervasive Computing Technologies for Healthcare, Tampere, 2008.

[17] J. Winkley and P. Jiang, "Adaptive probability scheme for behaviour monitoring of the elderly using a specialised ambient device," *Int. J. Mach. Learn. & Cyber*, in press.

[18] J. Winkley, P. Jiang, and W. Jiang, "Verity: an ambient assisted living platform," *IEEE Transactions on Consumer Electronics*, vol. 58, pp. 364-373, 2012.

[19] H. S. Shin, C. Lee, and M. Lee, "Adaptive threshold method for the peak detection of photoplethysmographic waveform," *Computers in Biology and Medicine*, vol. 39, pp. 1145-1152, 2009.

[20] T. L. Rusch, R. Sankar, and J. E. Scharf, "Signal processing methods for pulse oximetry," *Computers in Biology and Medicine*, vol. 26, pp. 143-159 1996.

[21] E. Sammer, *Hadoop operations*: O'Reilly Media, Inc., 2012.

[22] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260-269, 1967.

[23] N. Ye, "A markov chain model of temporal behavior for anomaly detection," presented at the IEEE Workshop on Information Assurance and Security, West Point, NY, 2000.

[24] J. Ying, T. Kirubarajan, K. R. Pattipati, and A. Patterson-Hine, "A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 30, pp. 463-473, 2000.

[25] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, pp. 164-171, 1970.

[26] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, pp. 401-409, 1969.

[27] J. Tenenbaum, V. d. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, 2000.

[28] J. A. Lee, A. Lendasse, and M. Verleysen, "Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis," *Neurocomputing*, vol. 57, pp. 49-76, 2004.

[29] P. Demartines and J. Herault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, pp. 148-154, 1997.

[30] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, pp. 37-66, 1991.

[31] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, pp. 21-27, 1967.

[32] J. Toyama, M. Kudo, and H. Imai, "Probably correct k-nearest neighbor search in high dimensions," *Pattern Recognition*, vol. 43, pp. 1361-1372, 2010.

[33] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," presented at the Proceedings of the thirtieth annual ACM symposium on Theory of computing, Dallas, Texas, United States, 1998.

[34] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors," *IEEE Signal Processing Magazine*, vol. 25, pp. 128 - 131, 2008.

[35] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality sensitive hashing scheme based on p-stable distributions," presented at the SCG '04 Proceedings of the twentieth annual symposium on Computational geometry NY, USA, 2004.

[36] O. Tene and J. Polonetsky, "Big Data for All: Privacy and User Control in the Age of Analytics," *Northwestern Journal of Technology and Intellectual Property*, vol. 11, pp. 239-273, 2013.

BIOGRAPHIES

Ping Jiang received his B.Eng, M.Eng and Ph.D degrees in Information and Control Engineering from Xi'an Jiaotong University, China, in 1985, 1988 and 1992 respectively. He is a Professor in Computer Science at the University of Hull, U.K, where he leads the research group of Intelligent Systems. His research interests include cyber-physical systems, wireless sensor and actuator networks, pattern recognition and machine learning, multi-agents, intelligent control and intelligent robots.

Jonathan Winkley graduated in 2009 with a B.Sc (Hons) in Robotics and Artificial Intelligence, and continued on to complete a Ph.D in the same field at the University of Bradford, U.K, in 2013. Currently an Animatronics and Electronics Engineer at The Seasonal Group, U.K, research interests include Robotics, Intelligent Programming and Animatronic Design.

Can Zhao graduated in 2011 with a B.Eng in Computer Science and Technology from Fujian University of Technology, China, and received a M.Eng in Control Science and Engineering at Tongji University, China, in 2014. He was a visiting student in Department of Computer Science at University of Hull, UK, in 2013. Research interests include artificial intelligence and wireless sensor networks.

Robert Munnoch graduated in 2007 with a M.Eng in Electrical and Electronic Engineering from Leeds University, and went on to work in industry as a systems integrator and control engineer for Cimlogic LTD. Currently studying for a Ph.D in Data Mining and Signal Analysis at the University of Hull, where research interests include Computer Vision, Data Mining and Robotics."

Geyong Min is a Professor of High Performance Computing and Networking in the Department of Mathematics and Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Next Generation Internet, Wireless Communications, Multimedia Systems, Information Security, High Performance Computing, Ubiquitous Computing, Modelling and Performance Engineering.

Laurence T. Yang (M'97) received the BE degree in computer science and technology from Tsinghua University, China, and the PhD degree in computer science from University of Victoria, Canada. He is a professor in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, big data, cyber-physical-social systems. His research has been supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation. He is a member of the IEEE.