ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Advancement of BLE Supported Healthcare System with Cloudlet

[1] Atrayee Gupta, [2] Nandini Mukherjee
[1][2] Department of Computer Science and Engineering Jadavpur University Kolkata

Abstract - **Previously, we defined Predictor virtual sensor as a precognitive sensor that can be utilized to track events in the priory, which is otherwise impossible for the resource-constrained physical sensor. In this paper, we explain how to embed intelligence with Bluetooth Low Energy supported health sensors, such as smart bands, using a PaaS based cloudlet architecture and Predictor virtual sensor. Here, we provide results using graphical analysis and mathematical models of case studies for precognition, which can be used for other kinds of sensors as well.**

Keywords: Internet of Things, Machine Learning, PaaS, Virtual Sensor

I. INTRODUCTION

Modern research focuses on improving lifestyle, using machine learning on sensors; those are used by people in daily life. Smart bands provide monitoring of our health using optical photoplethysmography (PPG) technology. By estimating the absorption rate of certain frequency of light by the red blood cells, scientists have evolved mechanisms to determine heart rate, SpO2, blood pressure, core body temperature, galvanic skin response (GSR), fatigue, respiratory rate, ecg and different other parameters of human body. These smart health devices use low power communication technology, BluetoothGatt low energy (BLE) [1] protocol, to transfer the vitals to a nearby Android system. These devices periodically transmit beacons, which are analyzed on android phone. However, these wearables or the android phones lack resources and support for further analysis, prediction or complex computation of the collected health signals. As the prediction of events requires learning the pattern of data, memorizing events, and sometimes complex processing, cloud computing becomes significant in this context. Therefore, sensor data are uploaded to cloud from android systems for further processing. However, in remote areas communicating to the back-end server in the cloud at real-time sometimes creates major issue. Prediction with missing sensor data in a remote health-care infrastructure becomes challenging. Moreover, these android applications do not provide support for embedding human intelligence with sensor data, which can be used as one component for next-generation health care system. Therefore in this paper, we introduce the concept of PaaS cloudlet architecture and *Predictor* virtual sensors [2] to predict events and estimate missing sensor data, minimizing the errors, at real-time.

The paper is organized as follows, Section II provides the architectural detail. Section III provides different *Predictor* models. Section IV discusses about results. Section V concludes the paper.

II. CLOUDLET ARCHITECTURE

We discuss the cloudlet architecture and explain the procedure involving capturing of sensor data from BLE enabled devices, transforming them in Android and pushing them in cloudlet for further processing.

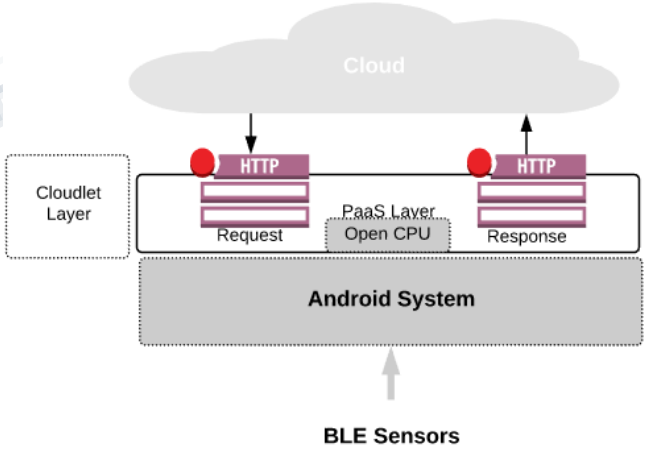


Fig. 1. Block Diagram for the Cloudlet Architecture

(a)Cloudlet and PaaS: Cloudlet introduced in paper [4], is formed by a single or a group of computing devices (such as personal computer), which lies at a one-hop distance from end user device and has the ability to communicate with the Internet. It minimizes the delay between edge devices [5], such as mobiles and wearables, and the remote cloud. It provides accessibility and real-

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

time computation facility at an intermediate level. In this research, we developed our own cloudlet architecture as shown in Figure 1 using OpenCpu [6], to manage sensor data in easier fashion. Novelty of this architecture is that it also provides flexibility to add computational libraries and relate continuous and categorical
variables for machine learning on sensor data. After receiving packets from a mobile device, proposed cloudlet processes the sensor data along with the other statistics submitted by the caregiver and acts as a docking station for sensors to the remote cloud users.

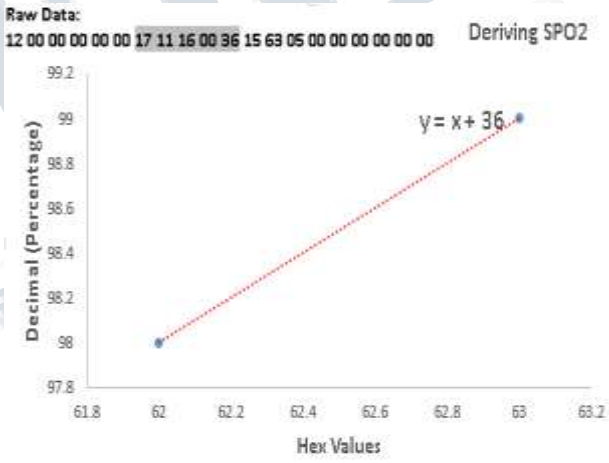
(a)Stages of Sensor Data Processing

Fig. 2. Stages of Processing of Sensor Data

As an intermediate node it not only facilitates communication but also provides a user the flexibility to reuse and even share the sensor data at same time. Here, sensor data, computational libraries and methods can be accessed via REST API calls from remote cloud end. Since cloudlet provides sufficient resource for processing and computation, most of the real-time processing of sensors can be done here. This facility is provided by the OpenCpu engine and machine learning capability of R in our cloudlet. Therefore, a physician can access this interface from the cloud, or the scientists as shown in Figure 2(a) can model the results.

Concept of virtual sensor (VS) is developed to abstract a physical sensor (PS) device at the cloud end and to achieve several functionalities including accumulating, sharing, and computing. These programmable software sensors can even be deployed to sense intelligently which is sometimes difficult for a physical sensor device. In our previous research work [2], we classified eight different types virtual sensors for different application requirements such as Singular, Accumulator [3] etc. Predictor VS allows us to estimate values from temperature, pulse etc. data from a patient and is effective in estimating missing sensor data caused by intermittent network connection or mobility. It is not only designed to guess the values from the predefined series of sensor data but also in advanced cases, may be used to track future events. We developed this Predictor VS on top of cloudlet architecture to aid in remote-health infrastructure.

(b)BLE Sensor Devices: BLE beacon carries the sensor data to an Android client in form of services and characteristics. For example, a blood pressure (BP) sensing service may contain several characteristics such as systolic BP, diastolic BP and a heart rate measurement. All these are packed inside individual characteristics under same service.



(b)Transformation of BLE Data

We enable notification or indication flag for those specific sensor characteristics with client configuration characteristic (CCCD) value inside a BLE packet, broadcast by the smart band. This ensures next sensor data will be available to us. We used heart rate sensors from MySignals cooking hacks and a smart band is shown in Figure 2(a). The entire process of capturing the sensor data from smart peripheral device, by a BLE supported android phone, starts with scanning and discovering the smart devices nearby, then enabling the sensors. This process is power consuming. Hence, we cannot afford high end exploratory data analysis with these vitals in the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

android phone. Therefore, we introduce a cloudlet with PaaS which can outsource the intelligence and gather advanced knowledge from different clients along with vitals collected from a person at real-time.

(c) Android System*:* After capturing BLE data packet shown in Figure 2(b), we transform the same using following methodology. We monitor each packet for patterns and carefully segregate the data according to the sensors. The health band uses heart rate, BP, SpO2, Steps count etc. We change one sensor data at a time, while keeping others fixed and derived a model for determining the values in readable format. We used BLE group specifications to determine manufacturer specific information, which is identified by universally unique identifier (UUID). Each UUID uniquely determines the characteristics, which holds the sensor data. We used the following algorithm to achieve the readable data for SpO2 and other sensors as well for smart band.

1) We search for UUID other than device information service
2) We search for characteristic that change with the change of sensor data.
3) We extract the string and split to search for known patterns
   a) We search pattern beginning with current date and time, and separate it out.
   b) We search for pattern of zeros, which signifies sensors which are switched off, and we separate it out.
   c) We extract the pattern which actually fluctuates with the sensor data.
   d) We match the extracted pattern with decimal values displayed in smart band.
      i) Fixed values against fixed reading
      ii) Inverted values (MSB or LSB) against sensor reading
      iii) We perform linear regression otherwise.
   e) We obtain decimal from these raw hex values
4) We check the validity of the result in next reading

MySignals provide software library, which allows us to instantiate the virtual sensors in Android phones, and use the result in human readable format. Once physical sensors are plugged, each of these VSs can be instantiated and results can be retrieved from the Hash Maps using their known characteristics UUID. In other case, result for SpO2, obtained using method 3(d)(iii) for a typical smart band, is shown in Figure 2(b), y-axis shows decimal value in %, x-axis shows raw hex data extracted from the string. We can observe, from the hex string grey area shows date and time. By observing few more strings, we extracted the desired sensor data and store them temporarily in Android cache, which are later pushed to cloudlet using XAMPP. In Cloudlet, we use OpenCpu and R libraries to model the pulse sensor data for prediction. We provide two use cases of Predictor VS in our cloudlet architecture in the following paragraphs.

A. Use Case I: Predictor for missing data analysis

Prediction reduces the number of transmissions in remote areas and provides efficient methodology to estimate data in case of mobility induced disconnections. Thus, estimating next data from the previous series is significant in these applications. Moreover, physical sensors generate random errors, thus predicting sensor data at cloud, also requires different error minimizing algorithms. Therefore, we design Predictor virtual sensor combining time variant ARIMA model and Kalman filters to predict any missing heart rate (resting) sensor data and minimize random error at real-time.

B. Use Case II: Predictor for Event tracking

In this case, Predictor has been designed to perform event tracking. We categorize two types of the events, such as regular events (E_{rg}) and rare events (E_{rr}) by monitoring of sleeping heart rate data. We coin the term of events based on the frequency of their occurrences. We used prediction using graphical analysis and categorization of events. It has been observed in the case study that Erg, such as lowest heart rate that is bound to happen in a period of total sleep cycle, can be predicted using categorization method by monitoring the HR pattern. Moreover, occurrence of Err such as fever can also be predicted using these methods on sleeping HR data.

III. PREDICTOR MODELS

PREDICTOR FOR MISSING DATA ANALYSIS

We designed the Predictor algorithm combining Kalman Filters (KF) and ARIMA (K_ARIMA) model. First we collected pulse of numerous persons from health sensor for a duration of two minutes per person and analyzed the same using time series (TS) analysis ARIMA. ARIMA(p,q,d) is a function of two series, one of which represents AR(p) and the other is MA(q). AR(p) is a function of its own past values and MA(q) TS is described as function of disturbances or time dependent linear function of past errors. The general equation of

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

ARIMA(p,q,d) or Box Jenkins method uses graphical analysis, correlogram analysis to find the parameters or components (p,q,d) of the TS. After that we do comparison among several ARIMA(p,q,d) models of the same series and choose the one with lowest AIC and BIC values which best fits the TS during training session. The sampling rate for the health physical sensors is kept one second. The time series equation for pulse of a person after finding the parameters of the model ARIMA(2, 2, 1) is given by:

$$\hat{x}_i = -0.31 + 0.923 * x_{i-1} + 0.359 * x_{i-2} - 0.282 * x_{i-3} + \varepsilon_i - 0.871 * \varepsilon_{i-1} - 0.532 * \varepsilon_{i-2} \quad (1)$$

Thus, equation (1) says, pulse value for i^{th} term ($\hat{x}_i$) is function of its lagged values and error terms (ε_i). However, for each person, the coefficients of the equation may vary.

Since, the ε_i is unknown for i^{th} term, which we are trying to predict, we break down the equation (1) into two steps using Kalman Filter algorithm. This approach also helps to reduce the random errors generated by the sensor and provides a true value of the pulse for i^{th} term. KF works in two stages **Prediction** and **Correction** as shown in Figure 3.

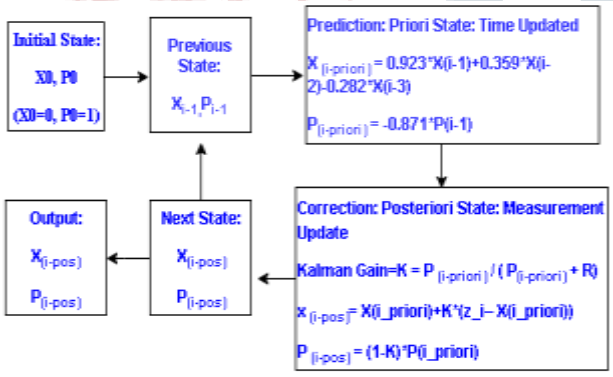


Fig. 3. Stages of K_ARIMA Algorithm

We break down the equation (1) into equation (2) and (3). $x_{(i-priori)}$ is same as $\hat{x}_i$ or pulse value for i^{th} term. $P_{(i\text{-priori})}$ is same as ε_i. In first step (Prediction), it tries to get an estimate of $x_{(i-priori)}$ in i^{th} term (known as Priori estimate). The suffix (i-priori), suggest, value of x, prior to correction.

$$x_{(i-priori)} = 0.923 * x_{i-1} + 0.359 * x_{i-2} - 0.282 * x_{i-3} \quad (2)$$

$$P_{(i\text{-priori})} = -0.871 * P_{(i-1)} \quad (3)$$

Equation (2) uses only lagged values and ignores the shift (-0.31). We ignore higher order terms to avoid stale data in equation (3). K_ARIMA buffers three measured values from sensor in z_i, initializes previous estimate $x_0 = 0$ and previous error $P_0 = 1$. In the next stage, Correction, the algorithm tries to adjust the value $x_{(i-pos)}$ in i^{th} state with actual measured value z_i. Since, outputs are corrected estimates of i^{th} state, hence suffixed as posterior to correction, $x_{(i-pos)}$ and $P_{(i-pos)}$. In this step equation (4), we find Kalman gain (K) as the ratio of errors, errors in priori estimates $P_{(i\text{-priori})}$ and errors in measurement process, R or the standard deviation of pulse. The K_ARIMA algorithm takes ten or more sensor readings to obtain R.

$$K = P_{(i\text{-priori})} / (P_{(i\text{-priori})} + R) \quad (4)$$

Next, we update the estimates of the state variable $x_{(i-pos)}$, and error matrix $P_{(i-pos)}$, using Kalman gain (K) obtained in equation 4, as follows:

$$x_{(i-pos)} = x_{(i-priori)} + K * (z_i - x_{(i-priori)}) \quad (5)$$

$$P_{(i-pos)} = (1-K) * P_{(i-priori)} \quad (6)$$

Utilizing the above equations (1-6) we derive the output variables (as estimates of missing sensor data) for each pulse sensor data z_i. This model differs for each person at real-time in cloudlet. Using proposed cloudlet architecture the complex calculation for missing data analysis is simplified with its open source libraries.

PREDICTOR FOR EVENT TRACKING

After collecting data from Android system, we use the following event tracking methods to determine lowest HR or E_{rg} and fever or E_{rr} in the cloudlet. In this research, we determine lowest sleeping HR for a person and term it as an event (E_{rg}). Since it varies with the person and is a continuous process, we use a smart band and graphical analysis tool for Predictor inside the cloudlet architecture. Average resting HR for a person lies between 60-100 bpm (or 50-90 bpm), measured when the person is awake,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

quiet and still. Sleeping HR drops further (40-50 bpm). We developed a learning model with sleeping HR data and found some interesting results. First, we determine optimum time-frame within which sleeping HR shows tendency to form a pattern. We say this time-frame as $T_{\min}$. If the same pattern repeats in next $T_{\min}$ we assume this is the learning model for sleeping HR for the person. We found that the learning model is specific for individual person and $T_{\min}$ lies between 7-8 days as illustrated in Section IV. We found that if we allot one sensor (smart band) for one person and keep it as fixed criteria, a model or pattern becomes visible after seven days. Even if the sensor is not accurate the pattern becomes visible.

Vitals are continuous parameters of human body and sensors measure them at intervals, which provide us discrete knowledge. However, once we recreate these signals by plotting these discrete values in a graph, we extract knowledge about regular events associated with the body. In this paper, we collected HR per hour and used graphical analysis to predict these events. We have noted from these analysis that these vitals show range of values, which forms regular pattern for the individual person. Therefore, instead of using the discrete value of lowest heart rate, we specify a time range or interval within which this event E_{rg} can manifest. Further, we assume that since lowest HR is dependent on several other unknown factors, it can be studied only by the behaviour of the continuous output stream monitored for several hours. Hence, we associate range or interval with the time frame (such as $T_{(i-1)}$ to $T_{(i+1)}$) illustrated in Section IV. Again, instead of specifying any discrete value, we specify range of values for lowest HR ($S_{(i-1)}$ to $S_{(i+1)}$) for the event E_{rg}. By doing so, we found sleeping HR shows a pattern, in which other events along with E_{rg} also becomes visible. We have studied that any abnormal conditions of the body becomes visible by studying sleeping HR pattern. In sensor terminology, they can be classified as events. The manifestation of these kinds of events can be predicted using continuous HR data. Therefore, the tendency of the graph to deviate from normal values indicate that other events are also associated with the regular events. These events, which fall outside the range of regular events, we term them as special or rare events E_{rr}, such as fever or any other abnormal conditions of the body. In this paper, we analyze sleeping HR data to predict the occurrence of such rare event E_{rr}. Any deviation (e.g sudden rise in HR) which does not follow a pattern (repeated for several days), is considered as noise in this case study. Then, we provide an optimum time frame of $T_{(rare-\min)}$ using sleeping HR graph, within which onset of E_{rr} can be predicted. This time usually varies from three-four day. Interestingly, it has been noted that the sleeping HR graph tend to maintain pattern at certain portions of the graph when E_{rr} is manifested and returns to normal pattern after E_{rr} is finished.

RESULTS AND DISCUSSIONS

Use Case I: Predictor for Missing Data

ARIMA	(2,3,1)	(2,2,2)	(2,2,1)	(2,2,0)	(1,1,0)	(1,1,2)
AIC	50.09	53.36	48.90	58.32	75.39	55.32
BIC	60.13	61.77	57.50	67.11	81.38	61.06

TABLE I: ARIMA Best Case Derivation for Pulse

Second-s	Pulse	Kalman	ARIMA	K_ARIMA	K_ARIMA constant
15	110	90.84	115.17	114.84	114.44
20	110	95.41	110.21	111.95	111.76
25	111	98.36	112.97	112.94	112.61
30	113	100.60	113.21	114.55	114.25
35	111	102.01	104.40	104.33	104.02

TABLE II: Comparison of estimates Kalman, ARIMA, K_ARIMA and K_ARIMA with constant with actual pulse

After comparison with several ARIMA, we found that ARIMA (2, 2, 1) is our best fit model for determining pulse as shown in Table I. Predictor Table II indicates the comparative results of the estimate of pulse variable using Kalman filter, ARIMA and K_ARIMA models. At t_i second, estimate $\hat{x}_i$ for i^{th} measurement, is tabulated below in Kalman, ARIMA and K_ ARIMA columns along with the actual measurement from the sensor. K_ARIMA with constant considers the shift in equation (1). Predictor in ARIMA model provides estimate closer to sensor values while Kalman filter provides measure close to the true value. Combined Algorithm K_ARIMA

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

(mean 102 bpm) provides better estimate of sensor values, reducing error at real-time. The convergence of the algorithm with sensor value is shown in Figure 4.

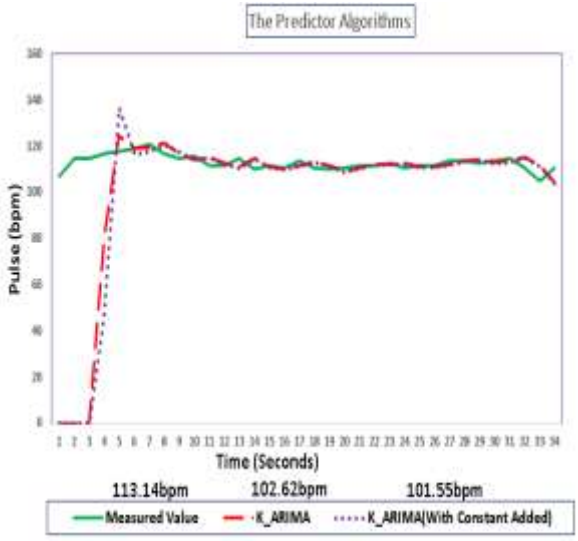


Fig.4. Convergence of K_ARIMA with Sensor Data

Use Case II: Predictor for event tracking

(a) Predictor for E_{rg}, regular event: In order to learn the model we plot the sleeping HR of the person, collected for several days, to obtain $T_{\min}$ period as shown in Figure 5(a) and Figure 5(b). Graphs tend to maintain a range of HR and manifested a low heart rate at time T_i, 4:00 A.M. We also found that the event E_{rg} may occur between a time-period ($T_{(i-1)}$ to $T_{(i+1)}$) as shown in Figure 6(a) and Figure 6(b).

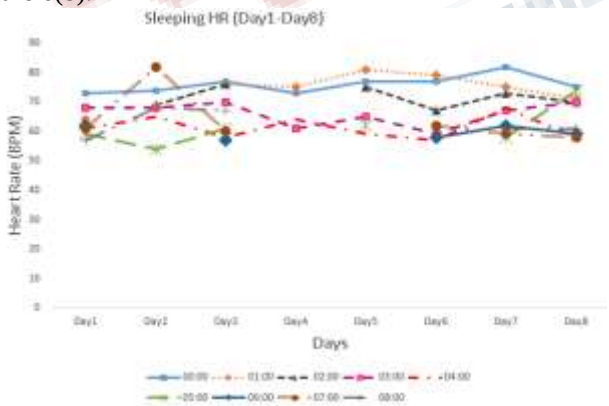


Fig .5(a). First 7 days of sleeping HR data

***(b) Predictor for E_{rr}, rare event*:** We found that the sleeping HR shows a tendency to form a peak before the onset of rare event, E_{rr}. Since the time frame for detection was also important, we evaluate an optimum time frame $T_{(rare-\min)}$ associated with the event E_{rr}. Figure 7(a), sleeping HR for day 12 shows tendency to deviate from normal pattern. On day 13, in Figure 7(b) shows peak due to E_{rr}.

Fig.5(b).Next 7 days of sleeping HR data

Fig.5. Predictor observations to obtain $T_{\min}$ period

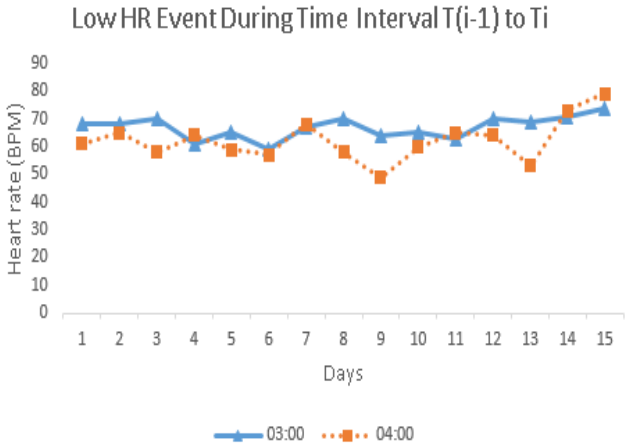


Fig.6(a). First Time Interval in which E_{rg} may occur

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

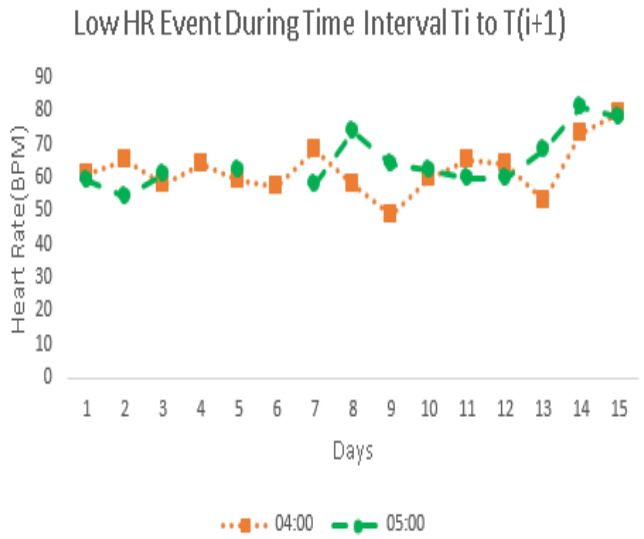


Fig.6(b). Second Time Interval in which E_{rg} may occur

Fig.6. Predictor observations for occurrence of Erg

The same pattern repeated on day 14 and day 15 in Figure 7(c) and Figure 7(d). The peak lowered at day 16 and day 17 during recovery phase and showed tendency to maintain normal pattern again, as revealed in Figure 7(e). Therefore, by continuous monitoring of sleeping HR, we are able perceive the onset of rare event associated with a person within an optimum time frame of $T_{(rare-min)}$ or 3-4 days.

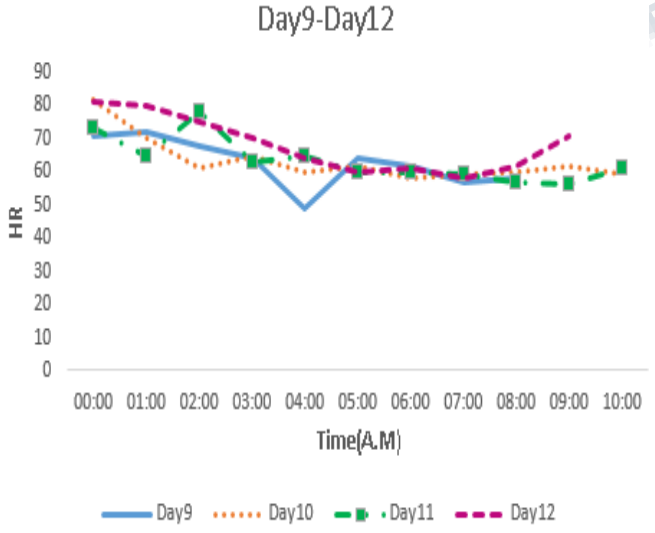


Fig.7.(a). Pattern of HR Before Onset of Event E_{rr}

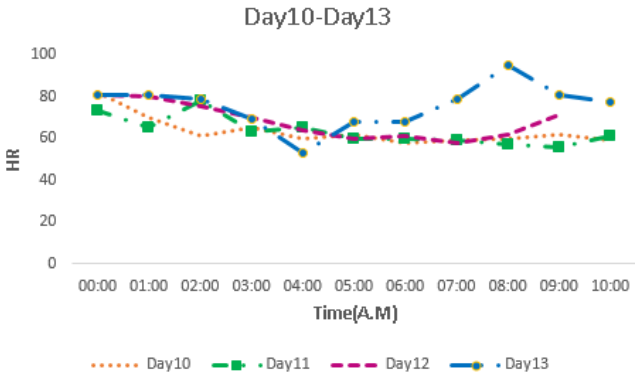


Fig.7(b). Event E_{rr} Occurred at Day 13

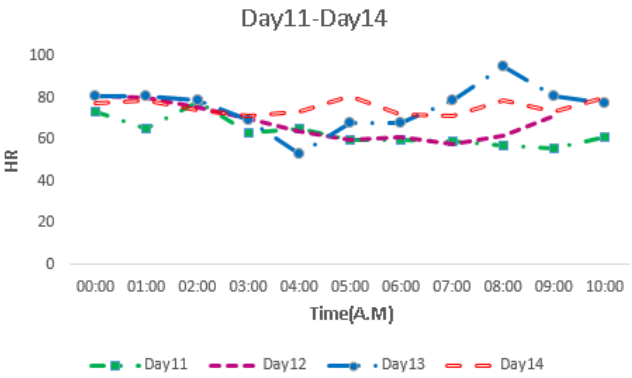


Fig.7(c). Repeat of Pattern for Day 13, 14

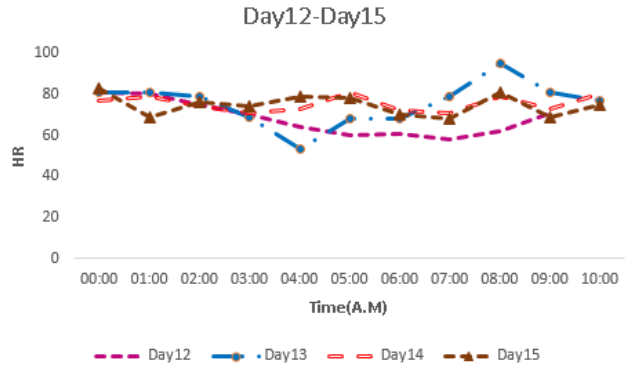


Fig.7(d). Repeat of Pattern for Day 13, 14, 15

CONCLUSION

In this paper, we developed the concept of Predictor virtual sensor to sense events ahead of its time using open source technology, cloudlet on non-invasive wearable sensors. Here, we showed how to implement cloudlet with BLE sensors and discussed several use cases to develop

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

prediction models using resting and sleeping HR. We developed a combined mathematical model K_ARIMA using time series analysis ARIMA and Kalman filter on resting HR to approximate next values of pulse minimizing error.

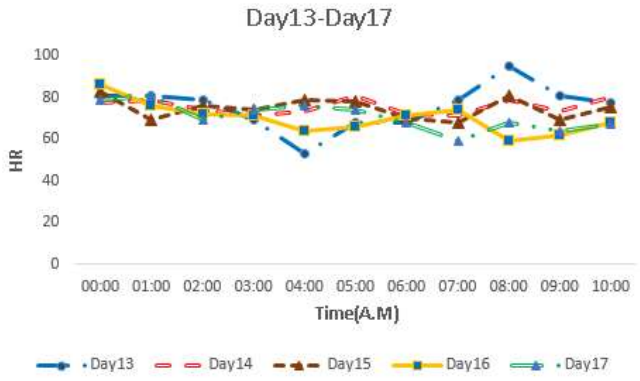


Fig.7 (e). Remission of the pattern after the termination of E_{rr}
Fig.7. Predictor for Event Tracking

In other case, we used graphical analysis for the Predictor on sleeping HR to detect the onset of any regular and rare event (for example, fever). In addition to remote health monitoring, this research work also provides the knowledge to build next-generation intelligent edge computing device using open source technology, sensors and human expertise.

ACKNOWLEDGMENT

This work is supported by Information Technology Research Academy (ITRA), Government of India under, TRA-Mobile grant [ITRA/15 (59) / Mobile/ RemoteHealth /01].

REFERENCES

[1] Shamsaa Hilal Al. Hosni, "Bluetooth Low Energy: A Survey", in Inter-national Journal of Computer Applications (0975 – 8887),vol.162, no 1, Mar. 2017.

[2]A. Gupta and N. Mukherjee, "Rationale behind the virtual sensors and their applications," in International Conference on Advances in Computing, Communications and Informatics (ICACCI),Jaipur,2016,pp.1608-1614, doi: 10.1109/ICACCI.2016.7732278

[3]A. Gupta and N. Mukherjee, "Implementation of virtual sensors for building a sensor-cloud environment," in 2016 8th International Conference on Communication Systems and Networks (COMSNETS), Jan2016, pp. 1–8.

[4]M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," in IEEE Pervasive Computing, vol. 8, no. 4, pp. 14-23, Oct.-Dec. 2009.
doi: 10.1109/MPRV.2009.82

[5]Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. 2015. "Edge-centric Computing: Vision and Challenges", in SIGCOMM Comput. Commun. Rev. 45, 5(September2015),37-42.DOI: https: //doi. org/10. 1145/ 2831347. 2831354

[6]Jeroen Ooms (2014), "The OpenCPU System: Towards a Universal Interface for Scientific Computing through Separation of Concerns" ,arXiv :1406. 4806, [stat.CO], URL http://arxiv.org/abs/1406.4806.