Three-Tiered Data Mining for Big Data Patterns of Wireless Sensor Networks in Medical and Healthcare Domains

Jong P. Yoon

MATH/CS Dept, Cybersecurity Program
Mercy College, Dobbs Ferry, NY 10522, USA
jyoon@mercy.edu

Abstract— **As smartphones become an emerging interface platform between humans and systems, they also enable wireless sensors to interface with host servers. As sensors monitor application domains and sensor data is frequently polled and transmitted to a host server, the server data will be a big volume, big variety and big velocity, which is the characteristic of big data. Mining patterns from big data is a very important and active research topic since it can be used to forecast and "nowcast" for any dynamisms in application domains. However, typical data mining algorithms are not successful yet due to the characteristics of big data. This paper describes three-tied data mining paradigm. Alongside the streamline of sensor data transmission, at the microcontroller tier, sensor data sets are mined to form patterns, at the smartphone tier, negative and positive patterns are grouped and verified, and finally at the host server tier, human expertise is associated with the patterns. The contribution includes 1) lowering data transmission by mining from the lower tiers, 2) mining time-critical data earlier than it would be done at the host server tier and 3) hence urgent responses can be made timely at the proper tier.**

Keywords- data mining; microcontroller;smartphone; wireless sensor networks; big data

I. INTRODUCTION

Big data is a large collection of complex data sets that are dynamically generated from various sources. It is therefore inefficient in processing or managing using traditional technologies. Data becomes large in volume if it is generated from various sensor networks. It should be transmitted in high speed if such data sets are processed centrally in a host. Wireless sensors monitor and poll various sensor data, which is then transmitted to a host server for further analysis and management. The data collected in a host is not only stored safely for archiving but it will also be analyzed to generate abstracted patterns or to associate meaning patterns for decision assistance.

In wireless sensor networks (WSNs), sensor nodes are deployed in unattended or less-attendable (nurses do attend and watch over patients 24/7) environments where wireless communication becomes more effective. Less-attendable hospital patient room environments may produce large quantities of data, which will be more dynamic if it is polled from various sensor nodes. Wireless patient monitoring devices are equipped with sensors capable of monitoring specific vital sign datasets [1,2,3]. For example in Figure 1, Electroencephalography (EEG) sensors are monitoring patient's brain activities, Electrocardiography (ECG) sensors are monitoring the function of patients' heart, a blood oxygenation sensor is equipped with IV (Intraventricular) for a patient while a motion detector monitors the patient and any infrared (IR) or ultrasonic sensors. Wireless sensors monitor patients, traditional medical equipment tools, and room environments [4]. Such wireless sensors as temperatures, pressure, moisture, chemical updates, images and audios, etc, can be deployed very easily and cost-effectively in medical treatment and healthcare, as they are running on microcontrollers such as Arduino [5] or its clones.

The radio signals polled by sensors are formed to be a sensor data. Sensor data polled from WSNs is then transmitted to the relevant medical practitioner and/or a host machine (see Table 1). A host machine may be a server or a cloud computing environment. Moreover it can often be a smartphone, which is widely and ubiquitously used especially in the medical and health communities. Sensor data or the outcome of sensor data processing will drive staff and doctors towards greater efficiency and quality of medical and health treatment. Healthcare and medical data will constitute big data [6,7], which will be more significantly big data if it is collected from WSNs.

One of the promising methods for sensor data analysis is data mining (DM). However, typical DM methods or DM technologies on integrated datasets [8] are not satisfactorily applied to the big data of sensor data sets. It is in part because sensor data of medical and health WSNs is

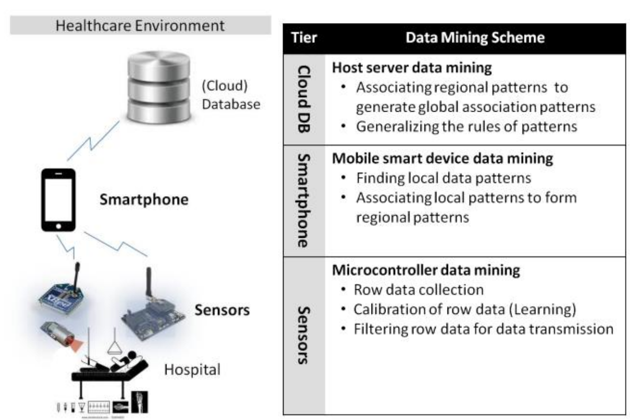


Figure 1. Sensor data streaming in hospital example, and data mining techniques along the sensor data streaming

 ISBN: 978-1-61208-280-6

streaming in dynamically from various sensor sources. The integrated volume of big data and the high-speed velocity of streaming WSNs make it inefficient for the DM to produce patterns that specific enough to correspond to the patients' responses in real time. If entire sensor data sets are integrated centrally in a host machine, sensor-sensitive local states are not efficiently taken into consideration and local exigent responses become impossible. In general, big data becomes difficult to process using on-hand database management tools or traditional data analytic applications. Simply speaking, automatically generated big data is almost impossible to be analyzed in servers' main memories only.

As such, this paper proposes a new wireless sensor healthcare approach to combine two architectural streams together between WSNs and DMs as shown in Figure 1. In WSNs on the left side, 1) multiple wireless sensor data is polled by microcontrollers, which will then transmit to 2) mobile smart devices and further to 3) host machines. In DM on the right side of the figure, multilevel DMs are proposed.

The contribution of the proposed research approach includes 1) lowering data volume transmitted from WSN node sensors and the host server where data is processed, 2) mining and analyzing sensor data early, in that the node specific actuation can be taken quick into consideration.

The remainder of this paper is organized as follows. Section 2 describes the sensors available for monitoring patients in healthcare and medical domains and some issues that we have encountered using them. Section 3 describes the three-tiered DM processes: 1) microcontroller DM at the sensors level, 2) mobile smart device DM at mobile phones level, and 3) server DM at a host or a cloud server level. Section 4 describes implementation details. Finally, Section 5 describes the conclusion and future work.

II. PRELIMINARIES

This section introduces sensors that can monitor patients for healthcare, microcontrollers that can operate, manage and wirelessly transmit sensor data, and smartphone devices.

A. Sensors

Many of sensors used in healthcare and medical services are traditional medical sensors: EEG, EMG, ECG, PPG, SpO2, etc. EEG (Electroencephalography) is a test that measures and records the electrical activity of brains. Neural oscillations are observed in EEG activities. EMG (Electomyography) measures the electrical activity produced by skeletal muscles, using a motor unit. When a motor unit fires, the impulse (i.e., action potential) is carried down the motor neuron to the muscle. ECG (a.k.a EKG, Electrocardiography) measures the electrical activity of the heart over a period of time. There are 12-lead ECG electrodes plus more to improve the sensitivity in detecting myocardial infarction involving territories not normally seen well. Blood pressure or more specifically Wearable blood pressure sensor [9] is a device that can monitor and measure and it could help diagnose hypertension and heart disease. This device uses pulse wave velocity, which allows blood pressure to be calculated by measuring the pulse at two points along an artery, and it monitor the hydrostatic pressure changes to prevent high blood pressure, which is a common risk factor for heart attacks. PPG (Photoplethysmography) measures pulse oximeters by cardiovasculary monitoring. SpO2 (Oxygen Saturation) measures the percentage of hemoglobin binding sites in the bloodstream occupied by oxygen.

Each such medical and healthcare monitoring equipment is equipped with the basic IR sensors. IR sensors include chemical sensors, acoustic and sound sensors, electric current and magnitude sensors, weather and moisture sensors, ionizing radiation and subatomic particles sensors, pressure sensors, distance and position sensors, force and level sensor, etc. Such IR sensors are loaded on microcontrollers and so they can be a wireless sensor as well as programmable.

B. Microcontrollers

A microcontroller is a single integrated circuit, which consists of a CPU, memory and programmable IO peripherals. It allows the firmware to handle interrupts in response of the events of sensors. There is a dedicated pulse width modulation (PWM) block, which makes it possible for the CPU to control power converters, resistive loads, motors, etc, without using lots of CPU resource. Sensor data can be periodically polled, but not stored in a microcontroller, since most microcontrollers have a limited storage (memory) space. Serial peripheral interface (SPI) and universal asynchronous receiver/transmitter (UART) can receive and transmit sensor data from/to external devices such smartphones.

For example, Arduino [5] consists of a simple open source hardware board with Atmel ARM or Atmel AVR, and a programming language (e.g., C and C++) compiler with a boot loader. A microcontroller board can support the aforementioned sensors as an Arduino shield. Example of communication shields includes wifi (IEEE 802.11) shields, Bluetooth (IEEE 802.15.1) shields, XBee shields (IEEE 802.15.4) [10], GPS shields, etc.

C. Smartphones

A mobile phone has also various sensors and communication components running on a mobile operating system such as Google's Android or Apple's iOS. The sensors equipped with most smartphones are proximity sensor, GPS sensor, cameras, accelerometer, etc. There are network protocols available, e.g., 3G/4G wireless communication, IEEE 802.11 wifi and 802.15.1 bluetooth, etc. As described in the previous subsection, since microcontrollers have insufficient storage spaces and they have the same network protocols, through IEEE 802.11 or 802.15 sensor data can be transmitted to smartphones.

ISBN: 978-1-61208-280-6

As such, the sensor data received from microcontrollers is stored in smartphones. Note that smartphones have an embedded database, more precisely a database library called SQLite. A smartphone has a storage capacity that is large enough to store sensor datasets in SQLite.

A rule of thumb is to assign one microcontroller per a patient or a patient room, where multiple wireless sensors deployed to multiple spots of patient body. Each medical or healthcare staff holds a mobile smart device, which polls sensor data from microcontrollers as they approach the microcontrollers.

III. THREE-TIERED DATA MINING

Since typical DM techniques are not satisfactorily applied to monitoring, mining and analysis in wireless healthcare sensor networks, this section describes three-tiered data mining. As shown in Figure 1, three DM models are available from the patient body, to the regional tier like each hospital floor, and the global tier like an entire hospital. The patterns mined from multiple microcontrollers at a patient body-tier are associated together on a smartphone at a regional tier, which will then be more generalized to form a discovery rules at the global tier server.

Note that this paper does not propose a new algorithm of data mining, but proposes a new paradigm of data mining in big data that is collected along the streamline of data transmission from wireless sensors to wireless smart devices and to host machines. Before proposing the three tiered-data mining paradigm, it is assumed that all sensor datasets are cleansed and trusted.

A. *Microcontroller Data Mining*

As described in Section 2, one or more sensors are plugged in on a microcontroller, which is called a WSN node. Multiple sensor data can be collected by a microcontroller at each every single polling [12]. For example in Figure 1, a WSN node may control EEG, ECG, moisture and pressure sensors. EEG and ECG respectively monitor the brain and heart activities of a patient, while the moisture sensor monitors the IV injection and the pressure sensor monitors the patient's bed. In this very common situation, two phases of data mining are proposed: Training and Calibration Phase and Pattern Transmission Phase.

1) Training and Calibration Phase: Each WSN node needs to be calibrated and the correlation of sensors needs to be identified by medical experts. Since it is unnecessary to poll data from all sensors at the same time in the same interval, a microcontroller should set a polling time for each sensor.

Suppose that four sensors, *s*1, *s*2, *s*3, *s*4 are deployed over a human body. Depending on the disease of patients, a different sensor set will be deployed. The sequence of such sensors is trained by calibration or determined by medical experts. A sequence of sensor data is written in regular expressions. A polling sequence (*s*1, (*s*2, *s*3)*, *s*4)* collects data from *s*1, and then *s*2 and *s*3 for multiple times, finally *s*4. This sequence is iterated.

For example, consider the following Arduino C programming of the nested iterations where four sensor datasets are polled in different frequency:

```
void loop() {
  sensorVal1 = analogRead(flexiForce);
  for (int i=1; i<=2; i++) {
    sensorVal2 = digitalRead(EEG);
    delay(100);
    sensorVal3 = digitalRead(ECG);
  }
  sensorVal4 = analogRead(liquidFlow);
}
```
(1)

The polling sequence of Expression (1) is (flexi force sensor, (EEG, ECG)*, liquid flow sensor)*, and more precisely, the iteration of the inner loop is 2. It means that between the flexi force and the liquid flow sensors are polled, a sequence of data polling from EEG and ECG occur twice.

2) Pattern Transmission Phase: From a sequence of polling sensor data discussed in the previous subsection, the pattern can be very easilly formed in the same expression of the polling sequence. The pattern obtained from the polling sequence is very similar to the pattern associated by A Priori association algorithms [11]. In the algorithm, an association pattern can obtained if the pattern has enough supports or frequency. In the same spirit, revisiting the sequence polling once more here, the pattern (s1 & s4) is formed since there exists a supportive pattern, (s2 & s3) with enough frequency. An example of pattern from Expression (1) is as follows:

```
(flex force = 0.1 mV) &
(liquid flow = 1000 nanoliter/min)

With the evidence of
(value of EEG = 60 microvolt/Hz) &
(value of ECG = 160 heartbeats)
```
(2)

Above Expression (2) is about a situation such that the flex force of patient's bed is almost negligible and the liquid flow rate of patient's IV is far greater than a nano liter per minute. This is indicative that a patient fell down from the bed and the IV injection is disconnected from the patient. This situation happens in many hospitals: at night a patient has abnormal pains, which is detected by EEG and ECG, and with no observation of medical staff, he or she moves from the bed and falls down. This risk of falling is a serious patient safety issue and a timely responsive actuator to a patient fall could help to minimize the negative repercussions of the fall.

The primary point of this research is that wireless sensor networks should be established and properly maintained and managed.

The beauty of this approach is that the pattern can be transmitted to a smartphone without waiting for more events

 ISBN: 978-1-61208-280-6

to count the enough frequency of *s*1 and *s*4. Only with the frequency of *s*2 and *s*3, e.g., in this case, after two occurences only, the pattern is quickly uploaded to a smartphone. If it is an urgent case, on the smartphone some additional alerts will be made, e.g., calling for a medical specialty. In a more traditional approach to patient care, the alerting of a patient urgently in need of attention may not be as timely.

B. *Mobile Smart Device Data Mining*

One example of widely used mobile smart devices is smartphones. Smartphones can communicate with wireless sensor networks. They receive sensor data from and/or transmit data or software to WSNs. Note that software packages can be transmitted to WSNs for several reasons [13], one of which is to upgrade. The goal of DM in mobile smart devices is to find correlations among wireless sensors, some from human bodies and others from hospital equipment and environments.

A smartphone can receive patterns, e.g., Expression (2), from one or more microcontrollers, each of which polls sensor data from one or more wireless sensors. Since each patient has different diseases and different symptoms, sensor data polled and correspondingly the patterns collected will be different. As such, at any point in time, it is likely there may be some patterns that are opposite of or conflicting with another. It may be in the case that (*s*1 & *s*4 & *s*5) and (*s*1 & *s*4 & ¬*s*5), where *s*# notes a sensor pattern and the symbol ¬ is a negation. If those two patterns are in the same database, due to conflict, no further reasoning is possible. However, they can be in two different databases for the DM purpose since they are about two different diseases, both will be an important mining factor. Moreover, for verification purposes in DMs, both conflict patterns can be considered together.

Suppose that there are patterns collected from microcontrollers: (*s*1 & *s*4 & *s*5), (*s*1 & *s*3 & ¬*s*5), (*s*4 & *s*5), (*s*3 & ¬*s*5), (*s*2 & *s*4 & *s*5), (*s*3 & ¬*s*5). We can split them into two groups: *positive example*, i.e., one with *s*5 and *negative example*, i.e., another with ¬*s*5. Thus, {*s*1 & *s*4, *s*4, *s*2 & *s*4} for *s*5 and {*s*1 & *s*3, *s*3, *s*3} for ¬*s*5 are obtained. The maximum common factor for each pattern set will be {*s*4} for *s*5 and {*s*3} for ¬*s*5.

However, if (*s*3 & *s*4 & ¬*s*5) is also included in the above scenario, the outcome will change. Since two pattern sets are {*s*1 & *s*4, *s*4, *s*2 & *s*4} for *s*5 and {*s*1 & *s*3, *s*3, *s*3, *s*3 & *s*4} for ¬*s*5, {*s*3} for ¬*s*5 is sound, but {*s*4} for *s*5 is not in the logic. The latter is untrue in the logic but may possibly be true in medicine; it should be notified to medical staff. The logical verification is not always true, but depends on real-world application domains. Therefore, the following definition is used in general for mobile smart device DM.

Definition 1 (*Positive and Negative Examples*) If there exists a pattern and its negation, called the pivot pattern, a set of patterns can be split into two groups. Positive example is a set of patterns that with the positive pivot pattern, and negative example is a set of patterns with negative pivot pattern. DM is performed in the positive example, while verification of DM in the negative example. (3)

Example 1: Consider the following patterns that are collected from microcontrollers.

```
(flex force = 0.1 mV) &
(liquid flow = 1000 nanoliter/min) &
(respiratory rate = 75 bpm)

(liquid flow = 1100 nanoliter/min) &
(respiratory rate = 72 bpm)

(heart beat rate = 160) &
(respiratory rate = 40 bpm)
```
(4)

It is known that normal healthy people have a respiratory rate between 10 and 45 breaths per minute. Hence, Expression (4) can be rewritten as follows:

```
(flex force = 0.1 mV) &
(liquid flow = 1000 nanoliter/min) &
¬(respiratory rate = OK)

(liquid flow = 1100 nanoliter/min) &
¬(respiratory rate = OK)

(heart beat rate = 160) &
(respiratory rate = OK)
```
(5)

According to Definition 1, the pivot pattern is `(respiratory rate = OK)` based on which Expression (5) can be split into positive example and negative example as follows:

```
(heart beat rate = 160) &
(respiratory rate = OK)
```
(*positive example*)

```
(flex force = 0.1 mV) &
(liquid flow = 1000 nanoliter/min) &
¬(respiratory rate = OK)

(liquid flow = 1100 nanoliter/min) &
¬(respiratory rate = OK)
```
(*negative example*)

The minimum pattern in the positive example is `(heart beat rate = 160)`, and the one in the negative example is `(liquid flow >= 1000 nanoliter/min) & (liquid flow <= 1100 nanoliter/min)`. Since any of these patterns can appear in the opposite example, the verification is done, and therefore

`(heart beat rate = 160)` for `(respiratory rate = OK)` (6)

`(liquid flow >= 1000` for `¬(respiratory` (7)

```
nanoliter/min)       &        rate = OK)
(liquid  flow  <=  1100
nanoliter/min)
```

Above Expressions (6) and (7) are obtained and sent (or texted) to appropriate medical and healthcare staff. If the mined patterns are meaningful, the staff may be able to take an action of treatment.

C. Host Server Data Mining

A host server is an enterprise system that holds not only wireless sensor-related data and patterns but also historical and statistical databases about hospitals.

The patterns formed in smartphones and taken care of by medical staff are uploaded to a server with their consequences. With patterns and consequences, a server can mine more useful and complete rules that may improve the overall healthcare program.

For example, back to the previous example as shown in Expressions (6) and (7), the following rule can be obtained:

```
(liquid flow >= 1000 nanoliter/min) &
(liquid flow <= 1100 nanoliter/min) &
¬(respiratory rate = OK) → emergency
treatment (E1, E2, E3)
                                              (8)
treatment (E1, E2, E3) → fail

treatment (E1, E3) → success
```

If a patient falls down from a bed, then a sequence of three treatments E1, E2 and E3 needs to be taken. Note that the treatment E# can be associated from the medical historical and statistical databases, which is omitted in this paper.

IV. Implementation Issues

This section describes some implementation issues along the streamline of sensor data transmission. Figure 2 shows three possible communication protocols: IEEE 802.11, 802.15.1 and 802.15.4. Smartphone Apps possessed by a medical staff initiate to find and connect Bluetooth devices, which run on microcontrollers (denoted as ① in Figure 2). One microcontroller can communicate with another (denoted as ② in Figure 2) or to the Smartphone Apps (denoted as ③ in Figure 2). Smartphone Apps then communicate with host servers (denoted as ④ in Figure 2).

The following code segments illustrate how an Android phone opens a communication session and communicates with Arduino BTshild. The key segments include the method findBT() and openBT(), then sending and receiving data.

```
void findBT() {
  mBTA = BluetoothAdapter.getDefaultAdapter();
  if(mBTA == null) {
     myLabel.setText("No BTA Available"); }
  if(!mBTA.isEnabled()) {
     Intent enableBT = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
     startActivityForResult(enableBT, 0);
  }
  Set<BluetoothDevice> pairedDevices =
mBTA.getBondedDevices();
  if(pairedDevices.size() > 0) {
   for(BluetoothDevice device : pairedDevices){
    if(device.getName().equals("myBT")){
       mmDevice = device;
       break; }
   }
  }
  myLabel.setText("BT Found …");
 }

void openBT() throws IOException {
  UUID uuid = UUID.fromString("00001101-0000-
1000-8000-00805f9b34fb"); // standard
  mmSocket =
mmDevice.createRfcommSocketToServiceRecord(uuid);
  mmSocket.connect();
  mmOutputStream = mmSocket.getOutputStream();
  mmInputStream = mmSocket.getInputStream();

  beginReceivingTransmittingData();
    myLabel.setText("Bluetooth Open…");
}
```

In response to the Android App's request, the hospital room's Arduino microcontroller takes actions: polling sensor data and transmitting the data to Android, as shown in the following code segment.

```
void loop() {
    while (bluetooth.available() == 0);
    fromPhone = (int)bluetooth.read();
    if (fromPhone >=10) {
      analogRead(sensor1);
      delay(1000);
    } else {
      analogRead(sensor2);
    }
   toXB2 = us.Ranging(CM);
    jServo.write(i);
    delay(100);
  }
  Serial.print(toXB2);
  if (Serial.available() > 1) {
    fromXB2 = (int)Serial.read();
    if(fromXB2>=10)
           digitalRead(sensor3);
    else digitalRead(sensor4);
    delay(5000);
  }
}
```

Note that the Arduino code also communicates with an XBee device (denoted as ② in Figure 2), which may sit on the same microcontroller or another separately.

The communication and data transmission paradigm illustrated in Figure 2 can be applied to various application domains. This section deploys 10 mobile WSN nodes in a closed space and illustrates how they identify the layout of the space. Each WSN node consists of an Arduino microcontroller, XBee transceiver antenna, a proximity sensor built on a servo motor and a time synchronizer as

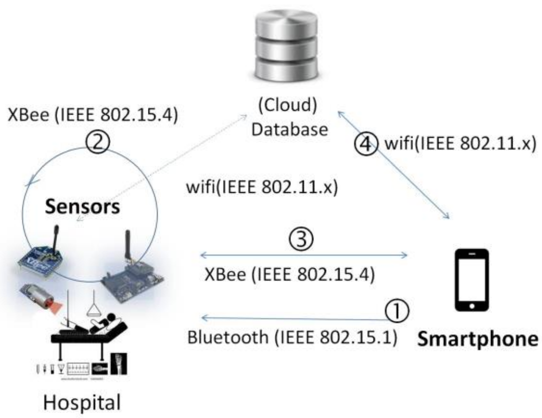
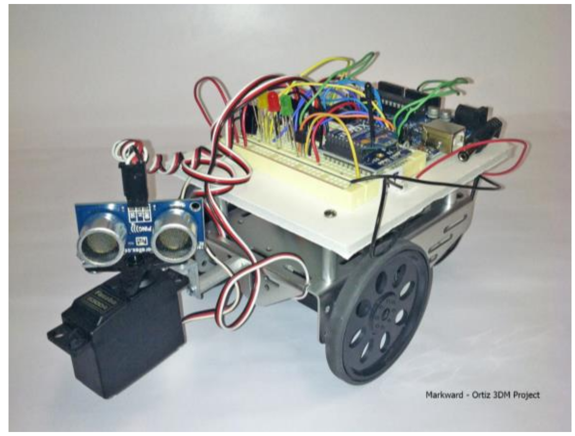
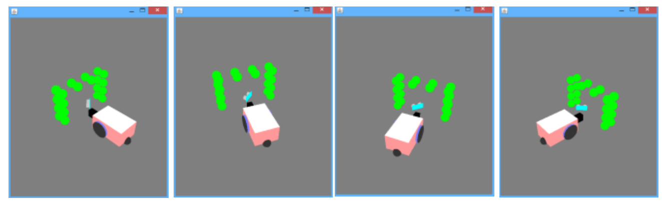


Figure 2. Communication among elements



(a) WSN node – real view



(b) Visualized views of the objects identified by each WSN node

Figure 3. An implementation of WSN

shown in Figure 3 (a). Each WSN node polls a sensor data about the objects that it may identify. Sensor data is transmitted to smartphones in the second tier, and finally reached at a host server in the third tier. Each WSN node and objects that are identified by the node are visualized at the server tier.

The sensor data polled from each WSN node is transmitted and finally reached at the server. Figure 3(b) illustrates the visualized views of the analyzed sensor data transmitted from four WSN nodes. The sensor data can be structured in 7 elements, (*id*, *x*, *y*, *z*, *a*, *d*, *t*), where *id* is the ID of a WSN node, *x, y* and *z* denote latitude, longitude and altitude, *a* is an angle, and *t* is the time of polling. The elements, *x*, *y* and *z* can be either an absolute geocode if GPS sensor is used, or otherwise relative coordinators. Given such sensor data, an object is recognized in the server. The data structure of such an object is structured in 4 elements, (*x*, *y*, *z*, *t*), where *x*, *y* and *z* are geocode as above and *t* is the time. The time *t* can be either synchronized if a time sync device is installed. Unless otherwise, it will be a local time. A typical approach is to transmit to the server all the sensor data polled. Note that in our experiment, each node polls and ships out the data at every 1000 msec. The smaller the time interval is set, the more the sensor data can be polled. The experiment presented in this paper does not use GPS devices and the system clocks in microcontrollers are not synchronized.

As discussed, filtering conditions are defined at each WSN node (microcontroller) or a smartphone. The filtering condition defined in each node filters the sensor data. The filter condition used in this experiment is: "if there are two data records polled by the same sensor point out the same object, only one data record is transmitted to smartphones." Another type of filtering condition is defined in smartphones. An example of such conditions is: "if there exist two data records transmitted from two different WSN nodes point to the same object, only one data record is transmitted to the server." This paper shows the experiment with the filtering conditions in WSN nodes only.

Another factor for monitoring and sensing objects is the specification of sensors. In our experiment, the factor is the range of proximity sensors. There are three ranges of distance measurement such as long-, medium- and short-distance measuring ranges. The long-distance measuring range of proximity sensors can cover beyond 1m, while the medium and short cover up to 100cm and 10cm, respectively. Sharp proximity of the short-distance measuring range sensor has been used in our experiment.

With the sensor conditions and the system configuration stated above, the preliminary outcome is shown in Table 1. It shows that the data transmission is reduced by 70% by the simple filtering condition at WSN node. Since there are numerously many sensors to be deployed in a real-world application (in military or in health care), more sensor data would be polled. Depending on the bandwidth of wireless networks, it may not be possible to transmit all the data polled. As such, the technique proposed in this paper makes it possible to reduce the sensor data transmission substantially and also to save the power required for WSN nodes.

Moreover, in an experiment with 10 WSN nodes, Table 1 shows that not all sensor data can be received at a higher tier. It is partly due to overwhelmingly big size of sensor data. Even in this case, however, the very same technique proposed here enables all essential data to reach at a higher tier in the proposed technique. Note that the table shows two cells with no numbers but N/A, meaning that not all the data records are reached at the server. In addition to the reduction of wireless sensor data reduction, our experiment shows that

Table 1. Data Transmission – Preliminary Result

	Data received in 1 min		Data received in 10 min	
	# of Records	Size (KB)	# of Records	Size (KB)
Sensor data from a WSN node without Filtering	34	278	345	2987
Sensor data from a WSN node with Filtering	8	64	57	419
Sensor data from 10 WSN nodes without Filtering	351	N/A	3891	N/A
Sensor data from 10 WSN nodes with Filtering	74	577	721	5548

there is no significant difference in patterns mined from between the sensor data with and the one without.

V. CONCLUSION

This paper described a three-tiered data mining from big data sets of wireless sensors deployed to medical and healthcare domains. From wireless sensors over hospital patients to a host server, three major agents are described: microcontrollers, smartphones and host machines. At each such device, data mining paradigm is investigated. Not only reducing sensor data by filtering, but also mining wireless data patterns is performed at the sensor data sets at a lower tier, i.e., at each WSN node.

The contribution of the proposed research approach is a transmission data reduction and timely data mining: 1) enabling WSNs to filter sensor data at the microcontroller tier and 2) reducing data transmission by transmitting time-critical data only to the smartphone, which then aggregate sensor data from various WSN nodes to conduct more efficient data mining.

ACKNOWLEDGMENT

This research is partially funded by Mercy College faculty development program. The author would like to thank James Markward and Juan Ortiz for the preliminary sensor robot implementation and Dr. Christopher Frenz and Dr. Petre Dini for their comments.

REFERENCES

[1] R. Naima and J. Canny, The Berkeley Tricorder: wireless health monitoring, Wireless Health ’10, October 2010, pp. 212-213.

[2] J. Woodbridge, A. Nahapetian, H. Noshad, M. Sarrafzadeh, and W. Kaiser, Wireless Health and the Smart Phone Conundrum, SIGBED Review 6(2), 11, 2009.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless Sensor Networks for Habitat Monitoring, ACM Int’l, Workshop on Wireless Sensor Networks and Applications, Septermber 2002.

[4] MIT Wireless Center, http://wireless.csail.mit.edu/index, Successful human tests for first wirelessly controlled drug-delivery chip, http://web.mit.edu/newsoffice/2012/wireless-drug-delivery-0216.html; MIT’s CSAIL launches new center to tackle the future of wireless and mobile technologies, http://web.mit.edu/newsoffice/2012/wireless-research-center-founded-1011.html retrieved in Dec., 2012.

[5] Arduino, http://www.arduino.cc/ retrieved in Aug., 2012.

[6] J. Manyika, et al, Big data: The next frontier for innovation, competition, and productivity, McKinsey Global Institute, http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation, retrieved in May 2011.

[7] J. O’Donoghue and J. Herbert, Data management within mHealth environments: Patient sensors, mobile devices, and databases, ACM Journal of Data and Information Quality, vol. 4, no. 1, Octover 2012, pp. 1-20.

[8] J. Lockhart, G. Weiss, J. Xue, S. Gallagher, A. Grosner, and T. Pulickal, Design considerations for the WISDM smart phone-based sensor mining architecture, SensorKDD’11, August 21, 2011, pp. 25-33.

[9] MIT, Wearable blood pressure sensor offers 24/7 continuous monitoring, http://web.mit.edu/newsoffice/2009/blood-pressure-tt0408.html, 2009.

[10] XBee, http://www.digi.com/xbee/ retrieved in January 2013.

[11] R. Agrawal and R. Srikant, Fast algorithms for mining association rules in large databases, Int’l Conf. on VLDB, 1994, pp. 487-499.

[12] V. Chandola, O. Omitamou, A. Ganguly, R. Vatsavai, N. Chawla, and J. Gaber, Knowledge discovery from sensor data, SIGKDD Explorations Newsletter, 12(2), 2011.

[13] S. Zhu, S. Setia, S. Jajodia, and P. Ning, Interleaved hop-by-hop authentication against false data injection attacks in sensor networks, Transactions on Sensor Network, 3(3), 2007.

 ISBN: 978-1-61208-280-6