Anomalous Data Detection in WBAN Measurements

M. Udin Harun Al Rasyid, Fajar Setiawan, Isbat Uzzin Nadhori, Amang Sudarsono, Niam Tamami
Politeknik Elektronika Negeri Surabaya (PENS), Surabaya, Indonesia
udinharun@pens.ac.id, ffsetiawan@yahoo.co.id, {isbat,amang,niam}@pens.ac.id

Abstract—**Wireless Sensor Networks (WSN) are vulnerable to numerous sensor error and inaccurate measurements. This vulnerability reduces the efficiency of many WSN application, such as healthcare in Wireless Body Area Network (WBAN). For example, faulty measurement from sensor gives a false alarm to healthcare personnel and lead to wrong patient's handling. Therefore, a system to differentiate between real medical condition and a false alarm will improve remote patient monitoring systems and other healthcare service using WBAN. In this paper, a novel approach is proposed to do anomaly detection using prediction method. The objective of this paper is to make a system which can differentiate between real medical conditions and false alarms. This system forecast a sensor value from historic values and compares it with actual data from real measurement. The difference is compared to a threshold value, which is dynamically adjusted. Then using majority voting algorithm to determine whether the data is an anomaly or not. The proposed approach has been applied to real datasets and compares the prediction methods and the size of the sliding window. Experimental results show the effectiveness of the system, indicated by high Detection Rate and low False Positive Rate.**

Keywords—Wireless Body Area Network; anomaly detection; prediction method; dynamic threshold; majority voting.

I. Introduction

Wireless Sensor Network (WSN) is currently used in various applications on many different domains, such as healthcare, farming, and environment monitoring applications. WSN can be used in the healthcare service domain to improve the quality of the service itself [1-3]. WSN for healthcare domain often called Wireless Body Area Network or WBAN. Sensors for WBAN are small-sized and capable of sensing human's physiologic parameters, such as heart rate (HR), blood pressure (ABP), pulse, respiratory rate (RESP), and oxygen Saturation (SpO_2). To do this, sensors must be attached to subject's body and continuously monitored from hospital or even home [4]. Those sensors' sensed data can be used in healthcare programs. For example, it can be used in a decision support system to decide patient's disease or to decide the right patient's handling.

The anomalous data on sensor's data can be caused by many factors, such as hardware faults, corrupted sensor, energy depletion, wrong calibration, electromagnetic interference, loose sensor's placement, malfunction, wrong application, etc. Some factors mentioned before can cause the inaccuracy of the data and in the end the application will give wrong information or often called false alarms [5].

Like what we described above, it is so important to be able to detect the data inaccuracy that sent from the sensor. This detection process must be done before it is used to give information for the patient so that it doesn't cause a false alarm. In data monitoring case, the number of the data will increase with the time's passing. So, if there is no fast processing for the data, the detection of inaccurate data becomes impossible.

Several methods have been proposed to do the detection of the data anomalies. The distributed method, the detection process performed on each sensor node but this method requires great resources, such as battery and memory on the sensor node. The other example is centralized method which the detection method is done at the application level; it requires that each sensor node must send the data continuously so that the process can run well.

There are few works have been done relating to this research about anomaly detection for WSN. Jha et al. [5] proposed a model of anomaly detection by several techniques. The proposed model using statistical technique to model the sensor data, Kernel Density Estimation to perform data mapping, weighting the data and outlier detection using Deviation Factor (DEVF) and Normalized Deviation Factor (NDEV). Researchers used a dataset obtained from Intel Berkeley Research Lab. That includes data from 54 sensor and run for 42 days. After the experiment, it can be concluded that the proposed algorithm provides high precision in detecting outliers from the dataset.

Fawzy et al. [6] proposed a novel in-network knowledge discovery approach which gives outlier detection and data clustering simultaneously. This research proposed a new-clustering based approach combined with k-nearest neighbor (k-NN) approach to classify outliers, i.e. data noise or an event. Their method consists of four steps. First step is pre-processing process; the clustering algorithm is applied on all data from sensor and group data into several clusters. Second step is outlier detection, for each formed cluster from first step; the outlier detection algorithm is applied to label each cluster as normal cluster or outlier. And then the third step is to classify the degree of outlier value whether it is caused by error or event. The last step is to calculate the trustfulness of each sensor node to increase our certainty in a specific node. Experiments on both synthetic and real datasets show that the proposed algorithm is better than other techniques in both effectiveness and efficiency.

978-1-5386-8079-7/18/$31.00 ©2018 IEEE

Jurdak et al. [7] did a survey to various strategies used for anomaly detection in WSN data. They categorize the strategies based on the type of the anomalies. The researchers create 3 categories for the anomalies, i.e. network anomaly, node anomaly, and data anomaly. For each category, there are many examples of anomaly cases. After defining the types of anomalies, the authors also conducted research on strategies that can be used in anomalies detection that have been developed by previous research. Various strategies are analyzed based on several categories, namely the concept, the status of the strategy, the type of the anomalies, the system architecture whether distributed or centralized, and its use. After conducting research on various types of anomalies and anomaly detection strategies, the researchers concluded the instructions for each design of anomaly detection system.

Salem et al. [8] proposed a light-weighted approach to do online faulty measurements detection by analyzing the data which collected from medical WBAN. The proposed approach performs sequential data analysis using a smart phone as processing station. The objective of this research is to raise alarm only when patients enter in an emergency situation and to discard false alarms caused by faulty measurements or broken sensors. The proposed approach is based on the Haar wavelet decomposition, non-seasonal Holt–Winters forecasting, and the Hampel filter for spatial analysis, and on for temporal analysis. The proposed approach is applied on real physiological dataset for experiment. The experimental results prove the effectiveness of the approach in achieving good detection accuracy with a low false alarm rate. The simplicity and the processing speed of the proposed framework make it useful and efficient for real time diagnosis.

In this research, we built a system that can do anomaly detection on data using centralized method. We use a centralized method because most of the healthcare applications are using a single node to sense the physiologic parameters. For the detection process, we used 2 prediction method from WEKA library. We used SMOReg method and Gaussian Process method. For the prediction process, we use past physiologic data collected from the WBAN sensors. This prediction method is combined with dynamic threshold which can give an error threshold value according to recent conditions of the patients. In the final stage there is a majority voting method that can distinguish between medical alarm and alarm anomalies. By using this method, we can quickly process anomaly detection with fast process and has a high degree of accuracy.

II. The System Design

A. System Architecture

A system to detect anomalies in WBAN measurement's data using prediction method is developed using java and WEKA library as shown in figure 1. We built our application using java environment for the programming language and using WEKA library to do the prediction process. We use two prediction algorithm, SMOReg and Gaussian Process. For the sensor data, we use several real physiological dataset. The dataset is an *.arff formatted file which contains the results of measurements on some physiological parameters on the human body. There are 5 physiological parameter used in this research, i.e. blood pressure, pulse, heart rate, respiration rate, and oxygen saturation. The format of the dataset must be adjusted to arff data format in order to be processed successfully. This dataset must be uploaded to the detection application. The application is a desktop-based application built using java programming language. Inside the application, each instance in the dataset will be opened and analyzed. The anomaly detection in this application is developed using combination of some methods that are assembled into a new algorithm, flowchart of this anomaly detection method is shown in figure 2. After the detection process is done, the application will show the index number of the instance which detected as an anomaly. The application will also make a report using chart. The chart will draw all the physiological parameter and give mark to all the anomaly instances.

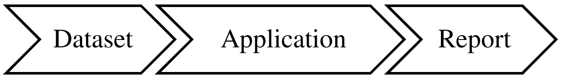


Figure 1. System Architecture

B. Anomaly Detection Approach

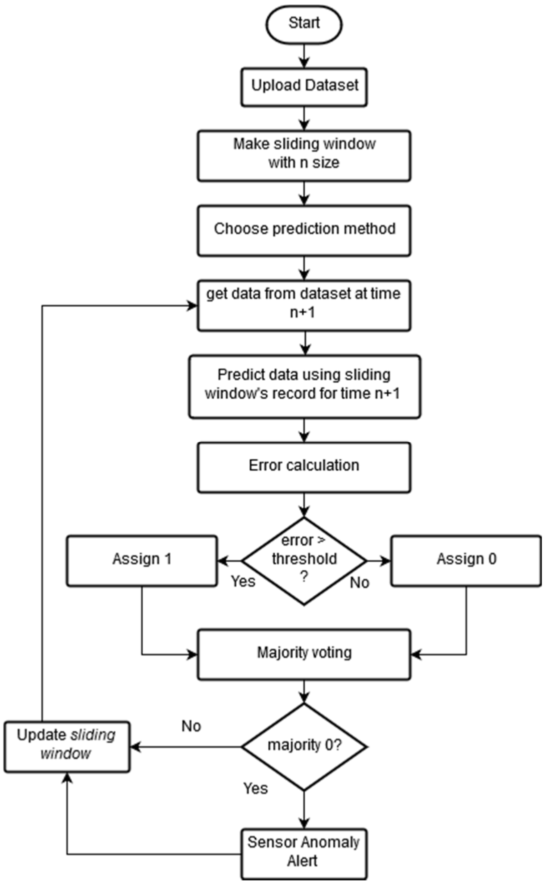


Figure 2. Anomaly detection flowchart

The proposed approach is based on three algorithm: SMOReg or Gaussian Process for the prediction of sensor value, Dynamic Threshold (DT) calculation for error calculation, and Majority Voting (MV) for decision on whether to generate alarm. Figure 2 shows the detection method we used in this research using these three algorithms. First, user must upload a dataset for the anomaly detection process. Then, the application will create a sliding window with inputted size. This sliding window is a subset of the uploaded dataset and will be used to create prediction model and dynamic threshold. The length of this sliding window can be adjusted from the application. The application will create a prediction model using selected prediction method and the data from sliding window. The next task is to get the physiological parameter value from the dataset. After that the application will perform a prediction using previously-made model. Once the data from both dataset and prediction method has been obtained, then these two value will be compared and the application will get the error value of each parameter. After getting the error value of each parameter, the next task is to compare whether the error values exceed the error tolerance/threshold. This tolerance value is obtained by calculating the standard deviation from the sliding window and multiplied with a constant. If the error value exceeds the threshold value, then the parameter will be labeled as an anomaly and given value 1, if not exceeds, it will be given value 0 and labeled as normal parameter. Finally, MV is used to detect false alarms and true medical condition. This process will compare the number of anomaly parameters with the normal parameters. If the parameter with anomaly labels more than half of the parameter count, the instance will be labeled as true medical condition. Instead, if the number of the anomaly parameters less than half of the parameters count, the instance will be labeled as anomaly. After that the data will be stored in a variable. Next is to make updates to the sliding window and headed back for the retrieval of data from the dataset. The process will continue until all existing data in the dataset has been analyzed.

C. SMO Regression

Sequential Minimal Optimization (SMO) Regression is an extension of Sequential Minimal Optimization for regression purpose [9] and an implementation for Support Vector Regression on WEKA. Suppose we have *j* data training [(x1,y2),....(xj,yj)] which can be the sensor measurement for patient's physiological parameters. A function *f(x)* with most error ( ) deviation from the actual training data. The errors are to be neglected as long as the values are less than . This is crucial as losing more than will deteriorate the system performance when dealing with medical data. This is the form of *f* for linear [7]:

$$f(x) = \langle w, w \rangle + b \text{ dengan } w \in X, b \in R \qquad (1)$$

Equation (1) means that it looks for a small w which can be ensured by minimizing the norm, i.e., ‖w‖2 = ‹w,w›. Here is the equation [9]:

$$\text{Minimize} \quad \frac{1}{2}\|w\|^2$$

Subject to:

$$y_i - \langle w, w \rangle - b \le \varepsilon$$
$$\langle w, w \rangle + b - y_i \le \varepsilon$$

(2)

The slack variables ξi, ξi* are introduced to deal with the optimization problem stated in Equation (2). This leads to the formulation stated in equation (3), here:

$$\text{Minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i, \xi_i^*)$$

Subject to:

$$\begin{cases} y_i - \langle w, w \rangle - b \le \varepsilon \\ \langle w, w \rangle + b - y_i \le \varepsilon \\ \xi_i, \xi_i^* \le \varepsilon \end{cases} \qquad (3)$$

C > 0 is a constant that determines error range.

D. Gaussian Process

Gaussian Process is a statistical model where observations occur in a continuous domain, e.g. time [10]. A Gaussian process is fully specified by a mean function $\mu(x)=E[Y_x]$ and a covariance function $k(x_i,x_j) = E[(Y_{xi} - \mu(x_i))\,(Y_{xj} - \mu(x_j))]$.

Gaussian for regression can be derived from Bayesian regression. For example, for every output y_i dependent on input x_i with this function:

$$y_i = f(x_i) + \varepsilon_i \qquad (4)$$

With ε_i as an error variable which spread with mean zero and variance σ^2, whereas x_i is an input vector at i with i=1,2,3….,n. If f functions assembled into a vector, they turn into $f=[f_1,f_2,...,f_n]$. So, according to Gaussian process it becomes:

$$f|X, \theta \sim N(0, K) \qquad (5)$$

With K as matrix n x n depend on X and θ, whereas θ is a parameter vector from covariance function. Each element (i,j) of the matrix K is k (xi,xj) hereinafter called covariance function. The equation (4) can be expressed in vector form as follows:

$$y = f + \varepsilon \qquad (6)$$

With *y* as result vector, *f* is vectors from regression functions, and ε is error variable vector.

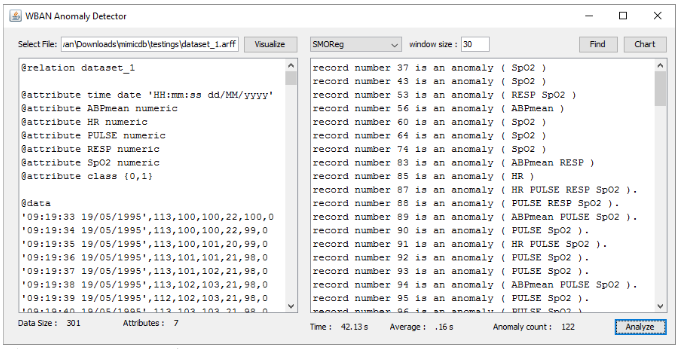
E. Dynamic Threshold

Human physiologic condition can vary for one subject to another based on some factors, i.e. age, physical condition, physiological condition, etc. Therefore, a static threshold or error tolerance can be used in every patient's case. Thus, it is important to have the ability to adapt the threshold value to one that reflects the actual overall physiological condition of the subject. Such a *dynamic threshold* value can be determined based on the subject's immediate past physiological data at the same time when prediction of a sensor value is performed based on the same historic dataset. Using standard deviation from patient's past physiological and multiplied by a constant, we have a dynamic threshold which can adjust to patient's immediate condition. The constant used in this dynamic threshold calculation we get from previous research which also discuss about dynamic threshold for anomaly detection process for WSN data [8]. In that study, the researcher used 1.95 as the constant for the dynamic threshold calculation. Incrementally updating the window with time provides a contextual viewpoint of the data that makes the threshold value dynamic and makes it more relevant to the data at various points in time. This dynamic threshold value is updated over time throughout the experiment.

F. *Majority voting*

Each subject is associated with a number of different types of sensors that measure different physiological parameters. Majority voting is performed for all the different physiological parameters calculated for an individual subject. The measured values of the physiological parameters are compared with the parameter values predicted for the corresponding sensors. Each physiological parameter is assigned a status of 1 or 0 indicating that the parameter is anomalous or normal. Assignments of all the physiological parameters of a subject are received and forwarded to voting. If the votes are greater than the average number of physiological parameters, the determination is made whether or not the sensor value is faulty based on the majority voting. We present an analysis to show the performance of the majority vote. Assume that the number of sensors is N and number of physiological parameters is n, where n ≥ N. Votes from the n are denoted as v(1), v(2), …, v(n) to assess the status (true alarm/ false alarm) of the system. The decision from the sensors is expressed as vi ϵ (0, 1) that is used for voting. If Y is greater than the average of the number of physiological parameters n, the majority vote determination is reported on true alarm or false alarm. It will be stated as true alarm if the outlier parameters are more than the average of the total number of parameters and it will be false alarm if the outlier parameters are less than the average of the total number of parameters.

G. *Visualization Design*



Select File: \an\Downloads\mimicdb\testings\dataset_1.arff

SMOReg   window size : 30

```
@relation dataset_1

@attribute time date 'HH:mm:ss dd/MM/yyyy'
@attribute ABPmean numeric
@attribute HR numeric
@attribute PULSE numeric
@attribute RESP numeric
@attribute SpO2 numeric
@attribute class {0,1}

@data
'09:19:33 19/05/1995',113,100,100,22,100,0
'09:19:34 19/05/1995',113,100,100,22,99,0
'09:19:35 19/05/1995',113,100,101,20,99,0
'09:19:36 19/05/1995',113,101,101,21,98,0
'09:19:37 19/05/1995',113,101,102,21,98,0
'09:19:38 19/05/1995',113,102,103,21,98,0
'09:19:39 19/05/1995',112,102,103,21,98,0
```

```
record number 37 is an anomaly ( SpO2 )
record number 43 is an anomaly ( SpO2 )
record number 53 is an anomaly ( RESP SpO2 )
record number 56 is an anomaly ( ABPmean )
record number 60 is an anomaly ( SpO2 )
record number 64 is an anomaly ( SpO2 )
record number 74 is an anomaly ( SpO2 )
record number 83 is an anomaly ( ABPmean RESP )
record number 85 is an anomaly ( HR )
record number 87 is an anomaly ( HR PULSE RESP SpO2 ).
record number 88 is an anomaly ( PULSE RESP SpO2 ).
record number 89 is an anomaly ( ABPmean PULSE SpO2 ).
record number 90 is an anomaly ( PULSE SpO2 ).
record number 91 is an anomaly ( HR PULSE SpO2 ).
record number 92 is an anomaly ( PULSE SpO2 ).
record number 93 is an anomaly ( PULSE SpO2 ).
record number 94 is an anomaly ( ABPmean PULSE SpO2 ).
record number 95 is an anomaly ( PULSE SpO2 ).
```

Data Size : 301   Attributes : 7

Time : 42.13 s   Average : .16 s   Anomaly count : 122

Figure 3. User Interface

Figure 3 shows the main design of the application. We use java's JFrame as the base of our application interface, using buttons with different function, text-area to display the input and output of the application, and some labels to give additional information for user. Our application has 2 main panel, i.e. left panel and right panel. On the left panel, there is a text-field which is used to upload dataset into the application. Under the text-field, there is a text-area which is used to display the contents of the dataset uploaded by previous feature. At the bottom of the left panel there are 2 labels showing the size and number of the attributes of the dataset that has been uploaded.

In the upper right pane there is a dropdown menu to select a prediction method to be used, a text-fields to enter the size of the sliding window, two buttons are the "Analyze" button and the "Chart" button. Analyze button is a major part of this research, this button is used to do the anomaly detection in the uploaded dataset. The results of this anomaly detection in the form of a list which row is an anomaly in the dataset. The results of this anomaly detection in the form of a list of any line that is an anomaly in the dataset that is uploaded. Chart button is used to create and open an HTML page which contains a chart of physiological parameters datasets and instances where there are anomalies in the dataset. And at the bottom there are three labels that will display the value when the anomaly detection process has been completed. The value on these labels indicate the detection time, the average detection of each data, and the number of data that is considered as an anomaly. The generated parameters graph as shown in figure 4.

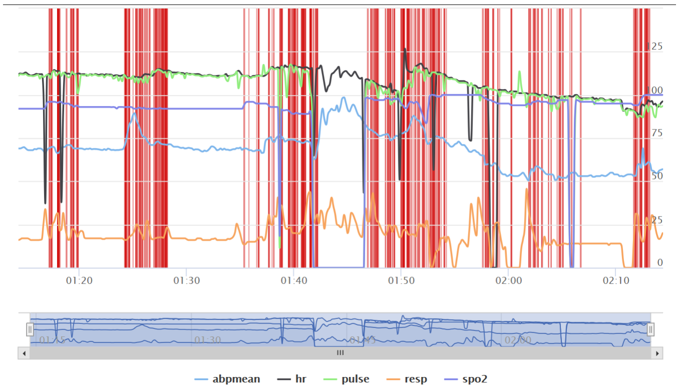


Figure 4. Generated graph

III. Performance Evaluation

In this performance evaluation section, we implemented the anomaly detection system using real physiological parameter datasets. Before we discuss about the evaluation of the detection system, we will describe something from the datasets we used until how we perform the detection experiments.

A. *Preparation*

Implementation and experiments conducted on real physiological datasets using the system which we developed in this research [11]. In this section, we will mention and describe datasets we used in the implementation process of this research. Therefore, we will explain how the author chose the datasets that will be used.

Multiple Intelligent Monitoring in Intensive Care (MIMICDB) database of PhysioNet [11] has 121 datasets of measurement results of human physiological parameters performed on real patients. Each dataset has different measurement's parameters. We choose datasets that has five parameters, namely blood pressure (ABPmean), heart rate (HR), pulse (PULSE), respiratory (Respiration), and oxygen saturation (SpO2). Besides choosing datasets based on the parameters, we also select datasets which have real anomaly so that we don't need to insert synthetic anomalies in datasets.

Each MIMICDB dataset has different length of measurement, ranging from 10 hours to 50 hours, which mean the dataset contains between 36000 to 180000 instances in each dataset. In this experiments, we cut the datasets to save time. We cut these datasets into small dataset. Each dataset we cut contains about 5 minutes data, or around 300 instances for each dataset. We created 4 datasets which contain both normal and anomalies dataset which vary in each dataset taken from some MIMICDB datasets.

Table 1. Created datasets

Name	**Source**	**Time**		**Instance count**
		From	**To**	
dataset_1	221n	19/05/1995 09:19:33	19/05/1995 09:24:40	301
dataset_2	054n	24/02/1995 21:02:00	24/02/1995 21:06:59	293
dataset_3	221n	19/05/1995 07:02:10	19/05/1995 07:07:10	304
dataset_4	414n	19/05/1995 13:39:57	19/05/1995 13:45:02	299

Table 1 above shows all datasets for the experiment process to test the system's performance. The table contains description about the source of the dataset, and the MIMICDB index number. Time column shows the time in which the dataset we taken from the original dataset. Then there is the instance count column which shows us the number of instances inside the dataset.

B. Testing Flow

There are two kind of experiments done to test the system's performance. Here is the types of the trials:

1. Changing prediction method.
2. Changing sliding window's size.

By conducting these trials, we aim to get two kinds of information, i.e. detection's accuracy and processing time. For the detection's accuracy evaluation, we used the confusion matrix. A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix [12].

Table 2. Confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Table 2 shows the confusion matrix for a two class classifier. Inside the matrix, there are 4 kind of conditions. Here is the 4 conditions from the confusion matrix:

1. True Positive (TP), is a condition where the system's output shows positive and the actual value is positive.
2. False Positive (FP), is a condition where the system's output shows positive but the actual value is negative.
3. True Negative (TN), is a condition where the system's output shows negative and the actual value is negative.
4. False Negative (FN), is a condition where the system's output is negative but the actual value is positive.

Using the 4 conditions, the Detection Rate (DR) can be calculated using equation 7 below.

$$DR = \frac{TP}{TP + FN} \qquad (7)$$

And the False Positive Rate (FPR) can be calculated using equation 8 below.

$$FPR = \frac{FP}{FP + TN} \qquad (8)$$

The higher the DR value and the lower the FPR value means the better the detection system.

C. Implementation result

The first implementation is conducted using SMOReg method for the prediction method and changes the sliding window's size (SW) used for the detection. Here is the first experiment's result

Table 3. SMOReg experiment's result (on %)

Dataset	**Detection's accuracy**					
	SW = 30		**SW = 45**		**SW = 60**	
	DR	**FPR**	**DR**	**FPR**	**DR**	**FPR**
dataset_1	100	4.6	**100**	**1.7**	**100**	**1.7**
dataset_2	100	4.5	**100**	**2.2**	**100**	**2.2**
dataset_3	100	4.9	**100**	**3.7**	100	4.5
dataset_4	95.7	10.9	100	8.0	**100**	**7.0**
Average	98.92	6.225	100	3.9	**100**	**3.8**

Table 3 shows the results of experiments conducted by changing the size of the sliding window and using SMOReg as a method to predict. Blue cells indicate the method with the best results in testing for a specific dataset. From the experiments have been conducted, it can be seen that by increasing the size of the sliding window on SMOReg method can minimize false positive ratio. As for the detection ratio, its value is always 100% in nearly all experiments. While on the detection of the dataset number 4 which has more variety of data, the detection rate of less than 100% when using size 30 and 100% when using size 45 and 60. The average of all experiments based on the size of the sliding window, it can be said that the size of the sliding window 60 is the best when using the method SMOReg with an average DR reaches 100% and an average of only 3.8% on FPR value.

Next is conducting experiments on Gaussian Process methods by changing the size of the sliding window used in the detection. The experimental results can be seen in Table 4.

Table 4. Gaussian process experiment's result (on %)

Dataset	**Detection's accuracy**					
	SW = 30		**SW = 45**		**SW = 60**	
	DR	**FPR**	**DR**	**FPR**	**DR**	**FPR**
dataset_1	100	2.9	100	1.7	**100**	**1.2**
dataset_2	100	12.0	100	2.6	**100**	**2.2**
dataset_3	100	4.1	100	3.7	**100**	**3.4**
dataset_4	95.7	8.3	100	6.3	**100**	**5.3**
Average	98.92	6.82	100	3.57	**100**	**3.02**

Table 4 is the result of experiments using Gaussian prediction method by changing the size of the sliding window. Blue cells indicate the size with the best test results in a particular dataset. As in experiments using SMOReg, this experiment has always achieved 100% detection rate in the three initial dataset, and the dataset to 4. It can be seen that by using a size 30, the results obtained are always the worst, while using size 60 result is always best. By using the average value as a reference, 60 is the best size to use Gaussian prediction method with the DR always reached 100% and the average FPR 3.02% and never reach a value of more than 5.5% in all sliding window.

From table 3 and 4 above, it can be seen that the setting of sliding window's size greatly affect the detection process. After experimenting with different prediction methods and datasets, it can be said that using Gaussian process as the prediction method with 60 data for sliding window gives the best result. This combination gives perfect value for the DR and false ratio of 3.02%.

In addition to performing experiments to test the level of accuracy in the detection, we also conducted an experiment to calculate the time spent processing applications in the conduct of each row of data. The scenario was the same as the previous experiment, by changing the prediction methods and the size of the sliding window. Furthermore, we note the results of the processing time in table 5 below.

Table 5. Processing time (on second)

Dataset	**Time**					
	30		**45**		**60**	
	SMO	**Gauss**	**SMO**	**Gauss**	**SMO**	**Gauss**
dataset_1	0.16	0.12	0.33	0.32	0.65	0.58
dataset_2	0.14	0.14	0.33	0.31	0.62	0.62
dataset_3	0.16	0.15	0.34	0.34	0.66	0.63
dataset_4	0.16	0.12	0.34	0.3	0.6	0.6
Average	0.15	**0.13**	0.34	**0.31**	0.64	**0.60**

Table 5 is a result of recording the processing time of each sample data at the dataset. It can be seen that the greater the size of the sliding window is used, then the time required to process the data becomes greater. The time needed by the two methods is almost the same for each size of its sliding window. When using size 30, the average time required by the program was never more than 0.17 seconds/instance. When using size 45, the longest time is approximately 0.35 seconds/instance. And when using a size 60, the processing time it takes about 0.6 seconds/instance. From the results of measurements of the average time required, it can be seen that the time required by the Gaussian method is always quicker than the method SMOReg despite the difference in time is less than 0.1 seconds.

IV. Conclusion

This paper explains the anomaly detection system that is used to distinguish anomaly data and not on human physiological parameters. The proposed detection system is to predict the value of a parameter using past data and compare it with the actual data. Then a dynamic threshold algorithm used to calculate the value limit error and continued on majority voting method to distinguish between anomalies on the sensor or actual medical condition. This method is implemented in an application built using the Java language using Weka library. We conducted an experiment by changing a few things on the applications and perform the registration detection accuracy, the detection rate (DR) and false positive rate (FPR). The implementation results show that using Gaussian prediction methods and the size of the sliding window 60 provides a good anomaly detection with an average ratio of DR and FPR respectively by 100% and 3.02%. The system can also be implemented on a real case using WBAN because it has a processing time of less than one second. For further research, application anomaly detection requires some updates such as the development of methods of dynamic threshold using machine learning methods and weighting on the training data in order to provide a basis for determining the anomaly is more appropriate.

Acknowledgement

This research was supported in part by Ministry of Research, Technology and Higher Education of Indonesia, under *Penelitian Terapan Unggulan Perguruan Tinggi* (PTUPT), Grant No. 0045/E3/LL/2018.

References

[1] T. Yilmaz, R. Foster, Y. Hao, "Detecting Vital Signs with WearableWireless Sensors," Sensors 2010, vol. 10, pp. 10837-10862, 2010.

[2] M.U.H.A. Rasyid, D. Prasetyo, I.U. Nadhori, A.H. Alasiry, "Mobile Monitoring of Muscular Strain Sensor Based on Wireless Body Area

Network", International Electronics Symposium (IES), September 29 - 30, 2015, Surabaya, Indonesia.

[3] M.U.H.A. Rasyid, W. Yuwono, S. Muharom, A.H. Alasiry, "Building Platform Application Big Sensor Data for e-Health Wireless Body Area Network", International Electronics Symposium (IES), September 29 - 30, 2016, Bali, Indonesia.

[4] O. Chipara, C. Lu, T. C. Bailey, G.-C. Roman, "Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit," Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems , pp. 155-168, 2010.

[5] V. Jha, S. K. Srivastava, "New and Fast Outlier Detection Scheme in WSN: NFODS-WSN," Journal of Global Research in Computer Science, vol. III, no. 1, pp. 42-44, 2012.

[6] A. Fawzy, H. M. Mokhtar, O. Hegazy, "Outliers detection and classification in wireless," Egyptian Informatics Journal, p. 157–164, 2013.

[7] R. Jurdak, X. R. Wang, O. Obst, P. Valencia, "Chapter 12 Wireless Sensor Network Anomalies: Diagnosis and Detection Strategies," A. Tolk and L.C. Jain (Eds.): Intelligence-Based Systems Engineering, pp. 309-325, 2011.

[8] O. Salem, Y. Liu, A. Mehaoua, R. Boutuba, "Online Anomaly Detection in Wireless Body Area Network for Reliable Healthcare Monitoring," IEEE Journal of Biomedical and Health Informatics, vol. XVIII, no. 5, pp. 1541-1551, 2014.

[9] A. Smola, B. Schölkopf, "A tutorial on support vector regression," Statistic and Computing, no. 14, pp. 199-222, 2004.

[10] C. E. Rasmussen, C. K. I. Williams, "Gaussian Process for Machine Learning," Massachusetts: MIT Press, 2006.

[11] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley," PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," Circulation, vol. 101, no. 23, pp. e215-e220, 13 June 2000.

[12] "Confusion Matrix," [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Accessed 20 June 2018].