World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:11, No:12, 2017

Implementation of an IoT Sensor Data Collection and Analysis Library

Jihyun Song, Kyeongjoo Kim, Minsoo Lee

***Abstract*—**Due to the development of information technology and wireless Internet technology, various data are being generated in various fields. These data are advantageous in that they provide real-time information to the users themselves. However, when the data are accumulated and analyzed, more various information can be extracted. In addition, development and dissemination of boards such as Arduino and Raspberry Pie have made it possible to easily test various sensors, and it is possible to collect sensor data directly by using database application tools such as MySQL. These directly collected data can be used for various research and can be useful as data for data mining. However, there are many difficulties in using the board to collect data, and there are many difficulties in using it when the user is not a computer programmer, or when using it for the first time. Even if data are collected, lack of expert knowledge or experience may cause difficulties in data analysis and visualization. In this paper, we aim to construct a library for sensor data collection and analysis to overcome these problems.

***Keywords*—**Clustering, data mining, DBSCAN, k-means, k-medoids, sensor data.

I. Introduction

A VAST amount of data has been generated by developing information technology and wireless internet. Most electronic devices include sensors which generate various data. These data can be used to provide useful information in real-time. Furthermore, we can obtain useful knowledge by analyzing accumulated sensor data. The weather center is a representative case. Development and dissemination of device control boards provide an environment to collect sensor data. By using a device control board and database tool, we can construct a database which includes various sensor data. These accumulated sensor data can be used for data mining. However, these tools are difficult and complicated to utilize by beginners. Furthermore, data analysis needs hard coding to implement various algorithms. To consider these issues, our goal is constructing a library to collect and analyze sensor data. By constructing modules for each sensor, we provide convenience for the data collection step.

For the data analysis step, we reduced the number of function arguments which are used in function calls for simplification. To construct a more understandable library, we defined function names as general words. The sensor data collection part is constructed based on the C programming language, and the sensor data analysis part is constructed using the R programming language [1].

Jihyun Song is with the Computer Science and Engineering Department, Ewha Womans University, Seoul, Korea (phone: +82-2-3277-2308; e-mail: ssongji7583@ ewhain.net).

Kyeongjoo Kim and Minsoo Lee are with the Computer Science and Engineering Department, Ewha Womans University, Seoul, Korea (e-mail: kjkimkr@ewhain.net, mlee@ewha.ac.kr).

II. Related Research

A. Sensor Data Processing

Sensor data processing requires streaming that sends data in real time and analyzes the data accumulated in real time. Streaming refers to the technique of playing back video, audio, and animation files on the Internet after they are stored in a data store, and playing them back during storage. For the streaming operation, it is necessary to collect data on the client side receiving the data, convert the data, and send it constantly to the application program. In addition, if the data reception rate is too fast to keep up with the processing speed of the data, a basic streaming structure is to place the buffer in the middle. Today, various streams of data and vast amounts of data are being generated, so various studies are under way to mining real-time data. Reference [2] used a clustering technique for streaming data processing and proposed an algorithm for obtaining high-quality clustering results for streaming data. Reference [3] selected a decision tree as one of the data mining techniques as a streaming data processing method. It presents an effective methodology for constructing a decision tree suitable for streaming data. Various studies for applying data mining technique to streaming data are being researched.

B. Sensor Data Analysis

Examples of sensor data include GPS data provided from an automobile, data generated from sensors (gyro sensor, ambient brightness sensor, GPS sensor, etc.) built in a smart phone, temperature and humidity sensor data. Using these data, new data can be extracted, and meaningful information extraction can be done to help many fields. Reference [4] proposed a study to measure human health using sensors. A method of designing and monitoring a sensor for measuring motion and heart rate is presented. Another example is analyzing sandstorms based on sensor data [5]. Through sensor data analysis, it is suggested that Aerosol and Brightness temperatures are related to presence of dust storms. These sensor data enable research in various fields, and various studies are still under way.

III. Library Implementation Using R

Our library, which is implemented for sensor data processing, has been implemented in four stages based on R programming. The library is divided into a sensor data preprocessing module, real-time event processing [6], database server, and data analysis module as shown in Fig 1.

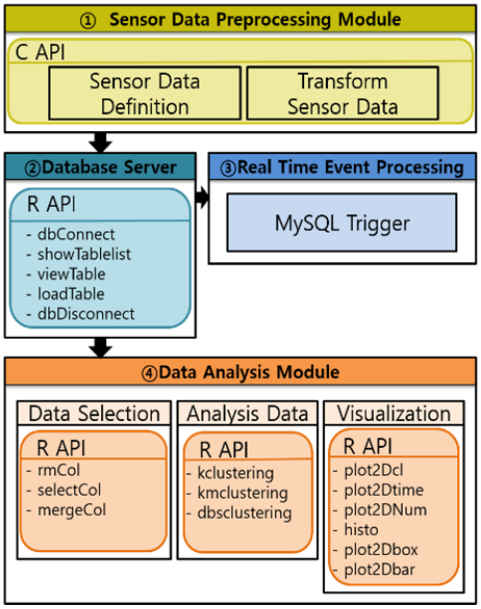

Fig. 1 Implemented library configuration

A. Sensor Data Preprocessing Module

The sensor data preprocessing module carries out sensor data definition process for setting basic environmental parameters for data communication between Raspberry pie and sensors. In order to collect sensor data, it is necessary to set the information to send and receive signals for each sensor and to convert the data according to the sensor. In this paper, we divided the sensor data definition part and the sensor data conversion part in the sensor data preprocessing module. In the sensor data definition part, it is possible to send and receive signals through the pin information setting for connection between the Raspberry pie and the sensor. Once the basic environment variables are set, they are converted into data types that can be understood by the user through the sensor data transformation process. In the sensor data conversion part, it plays the role of transmitting and receiving between the board and the sensor and proceeds to receive data from the sensor. Also, since the data received from the sensor are not immediately available, it includes an operation to convert it into usable data.

B. Sensor Data Storage

The database server is built on MySQL using the standard database query language SQL. Sensor data store that stores sensor data and external sensor data store that stores sensor data of web are constructed. The sensor data store is used to store sensor data transmitted from the Raspberry pie board. The external sensor data store is a storage area for storing sensor data that can be obtained from the web. Thus, the data stored in the database server is useful in the data analysis stage. In addition, since the database server and R are structured so as to be interlocked, analysis can be performed by immediately reflecting changes in the database.

C. Real-Time Event Handling

If it is necessary to process the data accumulated in real time, it is processed by the real time event processing. The real-time event processing step includes processing for transmission of erroneous values or additional processing operations according to a user's request. The processed real-time data are stored in the sensor data storage of the database server. The external sensor data store of the database server is a storage area for managing external data that can be obtained from the web. In this study, the trigger operates in a way to process the abnormal value in order to improve the quality of the sensor data when the abnormal value of the data is detected in the sensor data collection process. However, if one needs other features or wants to use additional features, one can change them with simple programming

D. Data Analysis Module

The data stored in the database server is moved to the data analysis stage to perform the analysis. The data analysis stage is divided into data selection, data analysis, and visualization. In the data selection step, the data in the database server are loaded, and only necessary parts of the loaded data can be selected and operated. In the data analysis phase, various clustering techniques are configured to be easy to use, and visualization steps are designed to visualize and display the results of data or analysis.

IV. Experimental Results

In this experiment, we compare and analyze the advantages of the proposed library over existing methods. In order to investigate what kind of analysis is possible using the currently constructed library, we analyzed the collected temperature and humidity data based on the proposed library and discussed the results of the analysis. We also compared and analyzed the number of command lines required to implement the implemented functions. In this paper, we compare the number of lines required for execution by dividing the case of using and not using the proposed library.

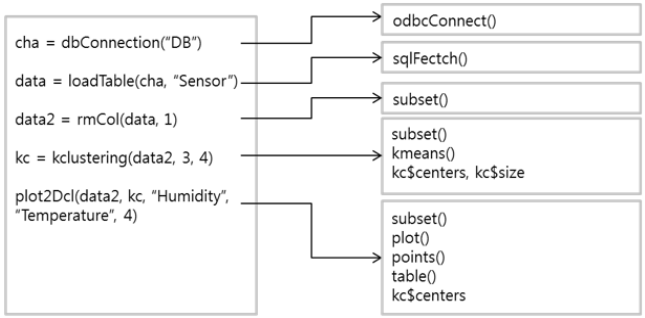

Fig. 2 Execution commands and functions and properties used by the library

Fig 2 describes information about the commands actually used to perform each process. In addition, we describe what functions are needed to execute each instruction. As shown in the figure, if we use functions provided by the library, we can manipulate, analyze and visualize data with five lines of commands. Therefore, it is very easy for the user to perform the analysis step.

A. Analyze Collected Sensor Data

The data used in this experiment are the measured values of room temperature and humidity change using a raspberry chip board. To collect the values, we used the constructed library and measured the environment by changing the indoor environment using a heating device to change the temperature and humidity. The temperature and humidity data used in the experiment can be regarded as the data indicating the change of the room temperature according to the heating mechanism.

In order to analyze temperature and humidity characteristics with time series, it was experimented to investigate which clustering [7], [8] technique shows the most appropriate analysis result value. The table below shows the results of applying the three clustering techniques provided by the library.

The data used in this experiment are the temperature and humidity change of the room measured with a raspberry chip in a cycle of 5 seconds. The temperature changes are in the range of 13 to 26 degrees, and the humidity is in the range of 31 to 37%. The total data size is 360 pieces. Table I shows the number of data for each level of the data divided by the three levels of the discomfort index.

TABLE I
THE SET OF TEMPERATURE AND HUMIDITY DATA USED IN THE EXPERIMENT

	Level 1	Level 2	Level 3
Data	79	166	115

Table II shows the results of analyzing the temperature and humidity data using the k-means [9] clustering technique. It can be confirmed that the analysis has been performed well for level 1, but it has failed to distinguish the other level 2 and level 3.

TABLE II
RESULTS OF K-MEANS CLUSTERING ANALYSIS

	Cluster 1	Cluster 2	Cluster 3
Level 1	67	12	0
Level 2	0	12	154
Level 3	0	0	115

Table III shows the results of clustering analysis of k-medoids [10] for temperature and humidity data. As it can be seen in the table, the clustering is quite good. The values were classified with an accuracy of about 95%.

TABLE III
RESULTS OF K-MEDOIDS CLUSTERING ANALYSIS

	Cluster 1	Cluster 2	Cluster 3
Level 1	74	5	0
Level 2	0	166	0
Level 3	0	0	115

Table IV shows the results of DBSCAN [11] clustering analysis for temperature and humidity data. As can be seen in the table, we can see that the clustering works well like k-medoids. In the table, cluster 0 refers to points that are located between the boundaries of the clusters and are not included in the cluster. In the case of DBSCAN, since the number of clusters is not specified, it can be seen that the results are shown by four clusters. DBSCAN classifies the values into about 90% accuracy.

TABLE IV
RESULTS OF K-MEDOIDS CLUSTERING ANALYSIS

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Level 1	18	61	0	0	0
Level 2	5	0	11	150	0
Level 3	0	0	0	0	115

When comparing three clustering methods, k-medoids and DBSCAN methods showed very good results and k-means clustering showed bad results. The k-means clustering method performs clustering based on the average value of the data, and thus fails to provide a good result in analyzing the data of dense temperature. On the other hand, in the case of k-medoids and DBSCAN, it is confirmed that it is a clustering method suitable for a given datum by defining a representative point and clustering nearby points with the starting point as a density-based approach. Since the performance of the cluster depends on the characteristics of the data, it is necessary to consider the characteristics of the data and the clustering technique together in the data analysis.

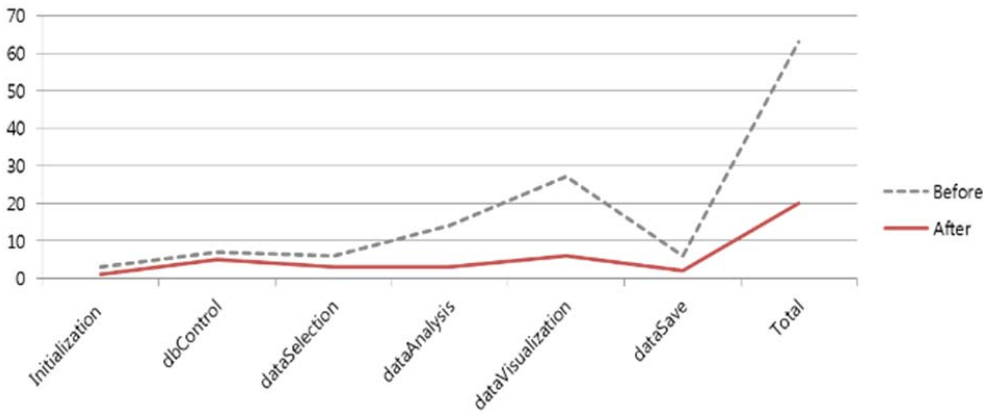

Fig. 3 Comparison of the number of execution lines

In this experiment, we also compared the number of command lines that the user actually has to enter in performing the same function. Fig. 3 shows the number of execution commands required to perform each function when the function provided by the implemented library is not used or used. As one can see in the figure, one can see that the number of lines required for all functions can be greatly reduced by using the proposed library. In the case of initialization and dbControl, there is no big difference because basically simple commands can be used to perform the functions. However, the required processes for dataAnalysis and dataVisualization are more complicated.

B. Analyze External Sensor Data

WIDSM [12] provides sensor data collected directly as public data. The data used in the experiment are the data of the acceleration change according to the movement of the user with the tracker data. Data are divided into six classes and has three types of characteristics. WIDSM collected data from 36 subjects and can be said to be a measure of the amount of change in acceleration for various motions. By analyzing these data, we can analyze the relationship between motion and acceleration. In this experiment, the WIDSM data were analyzed using the constructed sensor data analysis library. We extract N data for each class, analyze the extracted data through clustering, and analyze the data to analyze the analysis results.

A simple analysis and visualization work was performed based on the library that was constructed using the WIDSM sensor data provided on the web corresponding to external sensor data. In the WIDSM data, 3000 values were extracted for each type and analyzed based on a total of 18,000 data. As mentioned above, the six types consist of Downstairs, Jogging, Sitting, Standing, Upstairs, Walking.

TABLE V
RESULTS OF K-MEANS CLUSTERING ANALYSIS OF WIDSM DATA

	1	2	3	4	5	6
Downstairs	575	337	110	92	1849	37
Jogging	337	1256	1	657	135	614
Sitting	6	1	2878	0	115	0
Standing	1	0	0	0	2999	0
Upstairs	587	301	194	9	1909	0
Walking	855	648	68	210	1090	129

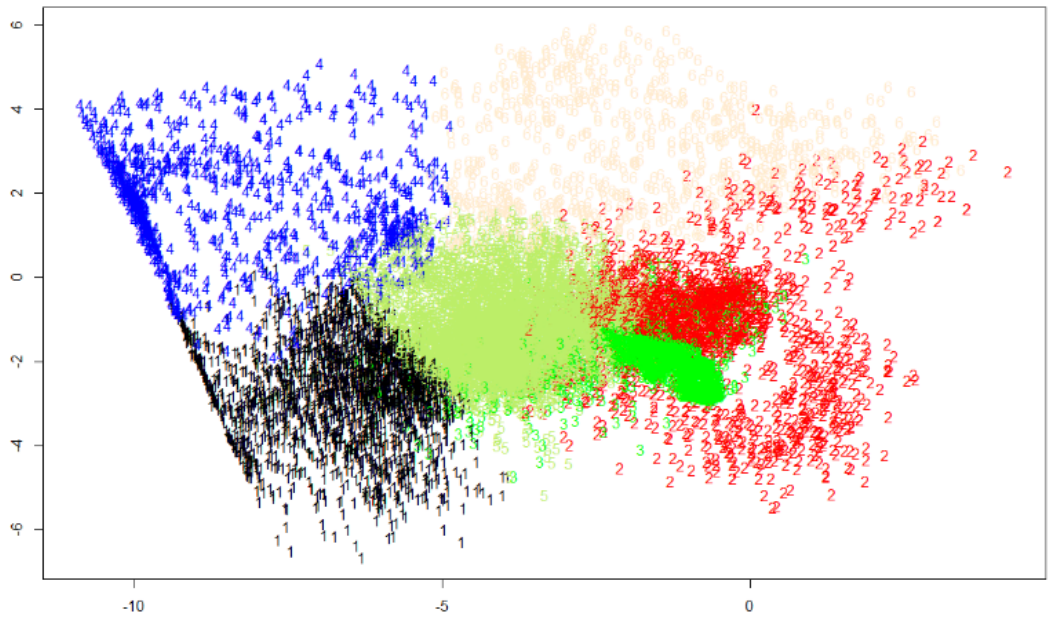

Fig. 4 Visualization of k-means clustering results of WIDSM data

Table V and Fig. 4 show the results of k-means clustering of WIDSM data. As can be seen in the tables and figures, it can be seen that sitting and standing are relatively well classified. However, we can see that other behaviors are not well categorized. This means that some behaviors have similar characteristics in X, Y, and Z axis acceleration. For example, it can be seen that there are acceleration values similar to the motion that occurs when a person climbs a stair and goes down a stair climb. Jogging and walking are distributed evenly in various clusters, which means that it is difficult to distinguish the behavior from the characteristics of acceleration. Therefore, additional attribute values are needed to distinguish these operations. For example, if we use sensor data such as heart rate and combine it with the above data, more sophisticated classification is possible. In this way, clustering can analyze which values are included in a similar cluster and which characteristics play an important role in distinguishing clusters. However, there are some cases where clustering techniques are not well categorized, so additional analysis is needed on what additional methods should be used to distinguish the various types.

V. CONCLUSION

In this study, we constructed a library for sensor data

collection and analysis for sensor data research. The goal of this research is to make users who do not have expertise in raspberry pies or R easy to use. To accomplish this goal, we implemented all the steps of data collection, analysis, and visualization at a minimum cost. In the case of sensor data collection, a module is built up for each sensor, and after simple pin setting, sensor data collection is made possible by selecting the desired sensor. In addition, sensor data stored in real time is managed through a database, and an exceptional data value is processed to further improve reliability of data. It also provides a method for analyzing the collected data and web sensor data using the R programming language. Data analysis and data visualization were modularized for each stage. Therefore, users can manipulate, analyze, and visualize data with a simple command. The number of execution commands is reduced by 68% compared to the case where the constructed library is not used. In addition, it provides convenience by providing overloading function for users to select various options.

The library provided in this paper is built for user convenience, but it does not yet contain many functions. The goal of future research is to build functions that are useful to users who need more functions. In addition, when defining a new function name in the data analysis module, we will consider naming the functions so that people in various fields intuitively understand the function of the function by name. The currently constructed library is based on function call. Therefore, it is true that the advantage in terms of speed is insufficient. In order to compensate for this, future research aims to analyze the R programming language for one step and change the structure inside the function or construct a new function. In terms of sensor data collection, we plan to implement a number of sensor data modules in addition to the sensors currently provided to enable various sensor data collection using the library. The future goal of this research is to construct a library that facilitates sensor data collection for users, provides easy access and quicker service for analysis and visualization.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2017R1D1A1B03034691).

REFERENCES

[1] R. M. Jayadeepa, M. S. Niveditha, “Computational approaches to screen candidate ligands with anti-Parkinson’s activity using R programming”, Current Topics in Medicinal Chemistry, Vol. 12, No. 16, pp. 1807-1814, August 2012.

[2] L. O’Callaghan, A. Meyerson, R. Motwani, N. Mishra, S. Guha, “Streaming-Data Algorithm for High-Quality Clustering”, International Conference on Data Engineering, pp. 685-694, February 2002.

[3] R. Jin, G. Agrawal, “Efficient decision tree construction on streaming data”, ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 571-576, 2003.

[4] A. Milenkovic, C. Otto, E. Jovanov, “Wireless sensor networks for personal health monitoring: Issues and an implementation”, Computer Communications, Vol. 29, Issues.13-14, pp. 2521-2533, August 2006.

[5] H. El-Askary, R. Gautam, R.P. Singh, M. Kafatos, “Dust storms detection over the Indo-Gangetic basin using multi sensor data”, Advances in Stalce Research, Vol. 37, Issue. 4, pp. 728-733, 2006.

[6] Khowaja, Ali Raza. "Process mining techniques: an application to time management." Ninth International Conference on Graphic and Image Processing. International Society for Optics and Photonics.H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[7] A. K. Jain, M. N. Murty, P. J. Flynn, “Data clustering: a review”, ACM Computing Surveys, Vol. 31, Issue. 3, pp. 264-323, September 1999.

[8] Andrew McCallum, Nigam Kamal, and Lyle H. Ungar. "Efficient clustering of high-dimensional data sets with application to reference matching." Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2000.

[9] Kanungo Tapas, et al. "An efficient k-means clustering algorithm: Analysis and implementation." IEEE transactions on pattern analysis and machine intelligence 24.7 (2002): 881-892.

[10] Hae-Sang Park, and Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering." Expert systems with applications 36.2 (2009): 3336-3341.

[11] Sanjay Chakraborty, and Naresh Kumar Nagwani. "Analysis and study of Incremental DBSCAN clustering algorithm." arXiv preprint arXiv:1406.4754 (2014).

[12] WISDM Public data, Online-Avaible: http://www.cis.fordham.edu/wisdm/.