Hindawi
Journal of Sensors
Volume 2018, Article ID 3265281, 13 pages
https://doi.org/10.1155/2018/3265281

Research Article

Application of a Mobile Chronic Disease Health-Care System for Hypertension Based on Big Data Platforms

Dingkun Li,[1] **Hyun Woo Park**,[1] **Erdenebileg Batbaatar**,[1] **Lkhagvadorj Munkhdalai**,[1] **Ibrahim Musa**,[1] **Meijing Li**,[2] **and Keun Ho Ryu**[1]

[1]*Database/Bioinformatics Lab, School of Electrical & Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea*
[2]*College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China*

Correspondence should be addressed to Keun Ho Ryu; khryu@chungbuk.ac.kr

Received 15 February 2018; Accepted 26 March 2018; Published 29 May 2018

Academic Editor: Ka L. Man

Hadoop is a globally famous framework for big data processing. Data mining (DM) is the key technique for the discovery of the useful information from massive datasets. In our work, we take advantage of both platforms to design a real-time and intelligent mobile health-care system for chronic disease detection based on IoT device data, government-provided public data and user input data. The purpose of our work is the provision of a practical assistant system for self-based patient health care, as well as the design of a complementary system for patient disease diagnosis. This system was only applied to hypertensive disease during the first research stage. Nevertheless, a detailed design, an implementation, a clear overview of the whole system, and a significant guide for further work are provided; the entire step-by-step procedure is depicted. The experiment results show a relatively high accuracy.

1. Introduction

Hypertension is a condition in which a person's blood pressure is above normal or optimal limit of 120 mmHg for systolic pressure and 80 mmHg for diastolic pressure. Increased blood pressure in the long term can lead to conditions that could threaten the health of the sufferer. Several conditions can cause disturbances in hypertensive cardiovascular organs such as stroke and heart failure, so sometimes mentioned that hypertension is a silent killer, because sufferers sometimes do not realize that he was exposed to conditions of hypertension [1]. The classification of blood pressure in adult divided into 4 classes which have been shown in Table 1.

Nevertheless, in our work, we treat prehypertension, HP stage 1, and HP stage 2 as hypertension with no difference.

Hadoop, which is based on MapReduce, has been one of the most important and popular techniques in the field of big data analysis during the last few years. Undoubtedly, it is the key technique for massive data analysis. Alternatively, Spark is a promising distributed framework that runs memory data on clusters at a speed that is considerably faster than that of Hadoop. Data mining (DM) is the key technique for the discovery of the useful information from well-processed data at the intersection of areas such as machine learning, statistics, and database systems. In the present work, the aim is the exploitation of the use of Hadoop, Spark, and DM techniques to provide a more powerful way of handling big data at high extents of speed, safety, and accuracy.

In recent years, the Hadoop framework has been widely used for the delivery of health care as a service [2]; moreover, a wide variety of organizations and researchers have used Hadoop for health-care services and clinical-research projects [3]. Taylor provided a detailed introduction on the use of Hadoop in bioinformatics [4], while Schatz developed an operations support system (OSS) package named CloudBurst

Table 1: Classification of Blood Pressure.

Classification	Systolic (mmHg)	Diastolic (mmHg)
Normal	<120	And <80
Prehypertension	120 ~ 139	Or 80 ~ 89
HP Stage 1	140 ~ 149	Or 90 ~ 99
HP Stage 2	≥150	Or ≥100

that provides an algorithmic parallelization model for which Hadoop MapReduce is used [5]. Indeed, the Hadoop framework has been employed in numerous important works to provide major contributions to the health-care field. The other big data processing framework, Spark, leverages a synergistic combination of the smartphone and the smartwatch in the monitoring of multidimensional symptoms such as facial tremors, dysfunctional speech, limb dyskinesia, and gait abnormalities [6].

Over many years, a large amount of health-care research work has been completed using DM techniques. In [7, 8], the authors used classification and regression techniques to predict conditions like cardiovascular disease and heart disease. In [9, 10], integrated DM techniques are provided for the detection of chronic and physical diseases. Further, a number of other research works, like [11, 12], used the advantages of DM to develop new methodologies and frameworks for health-care purposes.

The major goal of health-informatics research is the improvement of the quality and the cost of care that are provided to users, or the health-care output [13]. The purpose of the present work is the exploitation of Hadoop, Spark, and DM techniques for the design of a comprehensive, real-time, and intelligent mobile health-care system for chronic disease detection and prediction. The system is designed to provide an assistant system for self-based user health care, as well as a complementary system for the daily diagnostic work of doctors.

A series of challenges arise in the development of a big data-based health-care system. Firstly, it is extremely difficult to obtain high-quality and relevant medical data. One reason for this is that hospitals or the patients themselves are not willing to offer personal data for public research due to privacy policies. Another reason is the need to engage with a variety of data sources, such as the collection of data from hospitals, health-care centers, governments, laboratories, and the patients' families, which can cause serious missing-data problems. For instance, only the hospital-treatment data of patient A are available while the lifestyle (smoking, drinking, etc.) data are missing, and only the lifestyle data of patient B are available while the patient's treatment data are missing. The work of [14] confirms this varied-source characterization of health-care data collection and the complexity of different data forms. Secondly, data analysis is a challenging work. Even though a great quantity of research work has been completed to process and analyse data, a high-quality framework with highly precise predictive and analytic results is still mostly elusive [1]. Thirdly, the difficulty regarding the creation of a tool that can break the borders between the patient, health-care providers, and public health-care organizations to connect these parties in a practically meaningful manner is another obstacle [15].

The contributions of the present work are as follows: (1) exploration of the possibility of the utilization of Hadoop, Spark, and DM techniques in the work regarding health-care big data. (2) Depiction of a detailed step-by-step design of the health-care system for disease detection and prediction. (3) Provision of an overview for the next research stage and a guide for other similar systems. (4) Minimization of the monetary cost through the use of the Google Cloud services FCM and GCSql, which also guarantee real-time data transactions. A preliminary version of the present work has been reported in [15].

This paper is organized as follows: a description of the related work is provided in Section 2; an overview of the proposed system is introduced in Section 3; the design details are described in Section 4; the experiment results are described in Section 5; and Section 6 concludes this work and introduces future work.

2. Selection Techniques and Algorithms

This section briefly describes the related platforms, algorithms, and some of the key techniques that were used in the undertaking of the present work.

2.1. Hadoop, Spark, and Data Mining. Hadoop consists of the HDFS (Hadoop Distributed File System), HBase, and Hadoop MapReduce, making it very suitable for big data analyses [16]. As a 100% open-source framework, it has been widely used in almost every field for big data processing. In the last few years, Apache Spark [17] received great attention in the big data and data science fields, mainly because of its easier, friendlier application program interface (API) and an enhanced memory management compared with MapReduce; therefore, developers could concentrate on the data-computation logical operations rather than the background details of the computational execution.

It is difficult to find a coincident DM definition, but one of the widely accepted definitions states that DM is the process of discovering interesting patterns and knowledge from large amounts of data [18]. Its other close concept is called knowledge discovery in databases (KDD); DM is the analytical step of KDD. In this paper, a commonly used classification algorithm called C4.5 will be used for the disease-rule generation since it is simple, stable, and produces results of a relatively high accuracy.

2.2. C4.5. C4.5 is an algorithm that was developed by Ross Quinlan and is used to generate decision trees [18]. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees that are generated by C4.5 can be used for the purpose of classification, and for this reason, C4.5 is often referred to as a statistical classifier.

In general, the steps of the C4.5 algorithm for the building of decision trees are as follows: choose the attribute for the root node; create the branch for each value of that attribute; split the case according to the branches; and repeat

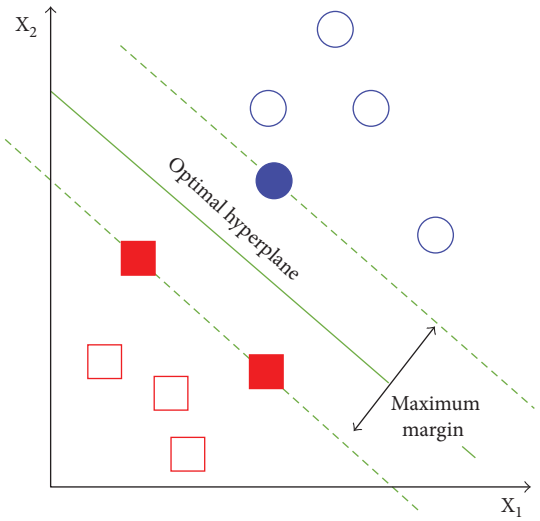


Figure 1: Maximum margin, the vectors on the dashed line are the support vectors.

the process for each branch until all of the branch cases are of the same class [18].

2.3. Support Vector Machine. Support vector machine (SVM) [19] has been used to select features and generate the classifier. For feature selection, this method is a backward sequential selection approach. One starts with all the features and removes one feature at a time until only r features are left. The operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. The basic concept is described using Figure 1.

The strategy ranks the features according to their influence on the decision hyperplane.

The optimal hyperplane is used to classify the data into different classes in two or more dimensionalities.

2.4. Hybrid Feature Selection Mechanism. Feature selection aims at finding the most relevant features of a problem domain. Primarily, there are two kinds of feature selection methods, filters and wrappers. The filters work fast but its result is not always satisfactory. While the wrappers guarantee good results, they are very slow when applied to wide feature sets which contain hundreds or even thousands of features. According to work [20], a hybrid feature selection mechanism takes advantage of both filter and wrapper feature selection methods is used to improve the computation speed and accuracy.

Inspired by [20], we developed our feature selection mechanism. The architecture is show in Figure 2.

3. Main Framework

An overview of the whole system is given in this section, and this is followed by a description of the implementation details in Section 4. The proposed system comprises four modules. The overview of the architecture of the entire system is depicted in Figure 3 [15]. The four modules in the figure

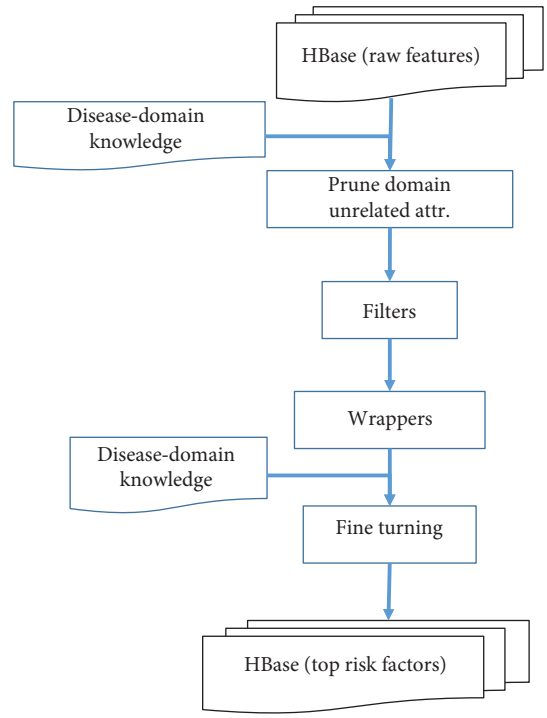


Figure 2: Risk factors (features) selection mechanism. We combined the expert domain knowledge for the purpose of pruning the unrelated attributes. Filter methods followed by wrapper methods are used select the features for disease rule generation. The evaluated result is stored in HBase.

are named as follows: (1) data collection module, (2) data storage module, (3) third-party-server (TPS) module, and (4) Cloud service module.

Module 1a is used to collect the streaming data and structured data by IoT devices such as Fitbit Charge 2, mobile phone sensors. 1b is used to import structured, semistructured, and unstructured data from various data sources like hospitals, governments, families, user inputs, and so on. Besides, we have developed a mobile app to collect user input data such as lifestyle and food intake data.

Module 2 is used to store the data in HBase collected by module 1. The data collected by the system is of three types: the structured, the semi-structured, and the unstructured data. Firstly, all these three kinds of data will be stored in HBase as it is quite suitable for mass data preprocessing and storage. Then this data should be converted into structured data for further processing.

Module 3 is used for the processing and analysis of the data based on the Hadoop/Spark cluster which is the key module of the whole system; all the data processing and analysis work will be done by this module. It is used for data statistical analysis, patient emergency detection, and disease prediction and detection. It also responses for message like data analysis results generation. These result will be sent to module 4.

Module 4 is used for the message dissemination. This model is implemented by using Google Cloud SQL (GCSql) and Google Firebase Cloud Messaging (GFCM) services. When receiving the requests from the TPS, Cloud model

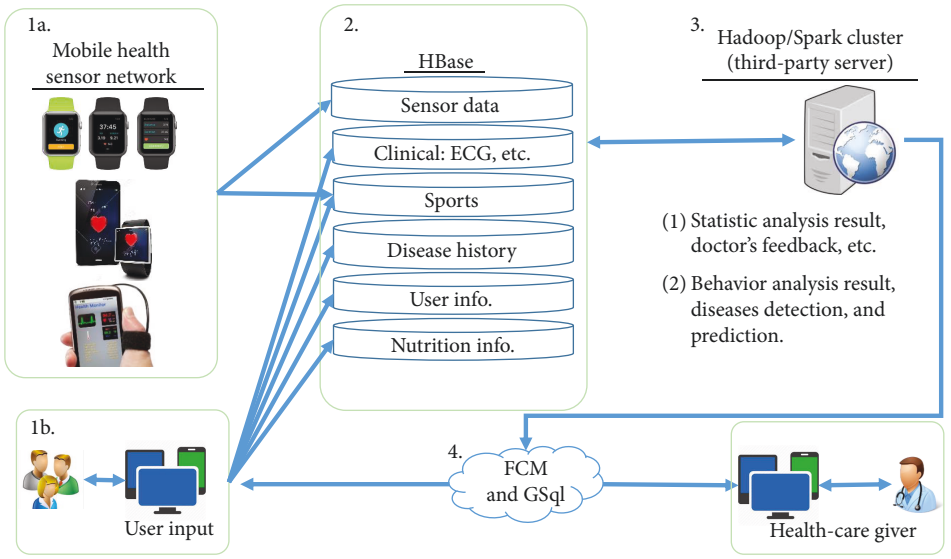


Figure 3: Systemic architecture: the four modules are marked 1 (1a and 1b), 2, 3, and 4.

responses immediately according to these requests, stores data, or sends data to the devices registered to it.

Further details have been given in a previous work of the authors of the present study [15].

4. System Implementation Details

In this section, descriptions of the systemic data flow, the data storage and processing, and the disease detection and prediction based on large medical datasets are provided.

4.1. Data Collection, Preprocessing, and Storage. To obtain high-quality structured datasets, database processing, natural language processing (NLP), and image-processing techniques are combined with the DM data-preprocessing techniques that are used by the TPS to process the different kinds of data (structured, semistructured, and unstructured), and the data are then transformed into a structured data record. The result is then stored in the HBase.

(1) For the structured dataset (mostly imported from the other public Web services) that includes patient information, prescriptions, and disease histories, it is relatively easier to import the data from the rational DB to the HBase using Sqoop [21].

(2) For the semistructured dataset, which includes HTML, XML, and Json documents, the TPS will design row keys like the d001 for the HBase table, including its document-information column value together with its family map that is called "column family" (it comprises the document timestamps of the HBase), as shown in Figure 4. The semistructured data will be converted to the structured data in the HBase, as shown in Figure 5.

(3) For the unstructured data, like clinic notes and the stream data from mobile sensors, they will be managed by the system in a particular way. The clinic notes contain a lot of the textual information, [22] providing an efficient way to convert this data into structured data through the use of NLP techniques, text-mining algorithms, and the MapReduce framework. The same strategy is used in the proposed system to deal with this problem, and the procedure is shown in Figure 6.

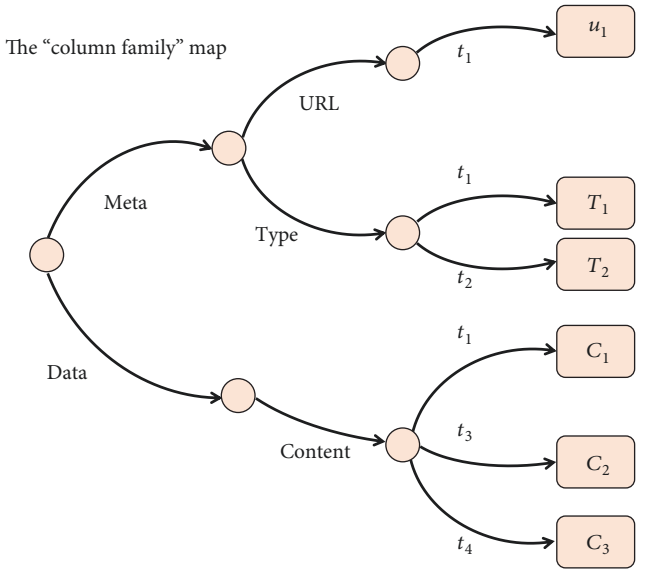


Figure 4: Column family map. The meta information and data content can be depicted by this map with their timestamp information. Where u_i stands for document URL, T_i stands for document type, and C_i stands for document content.

The stream data are handled using Apache Spark [23] techniques; the basic procedure is shown in Figure 7. Finally, the output will be stored in the HBase. After the preprocessing step, all kinds of data will be converted into structured data and stored in distributed HBase regional servers for further processing.

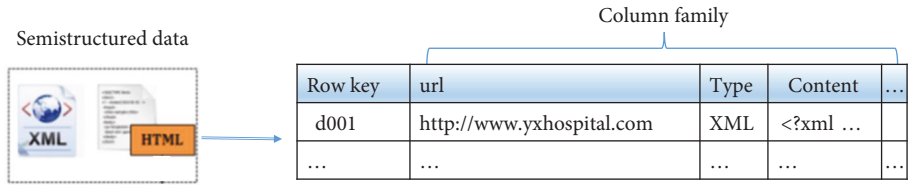


Figure 5: HBase example for semistructured data storage. Semistructured data (such as XML and HTML files) will be converted to structured data which has the similar structure as relational data base table.

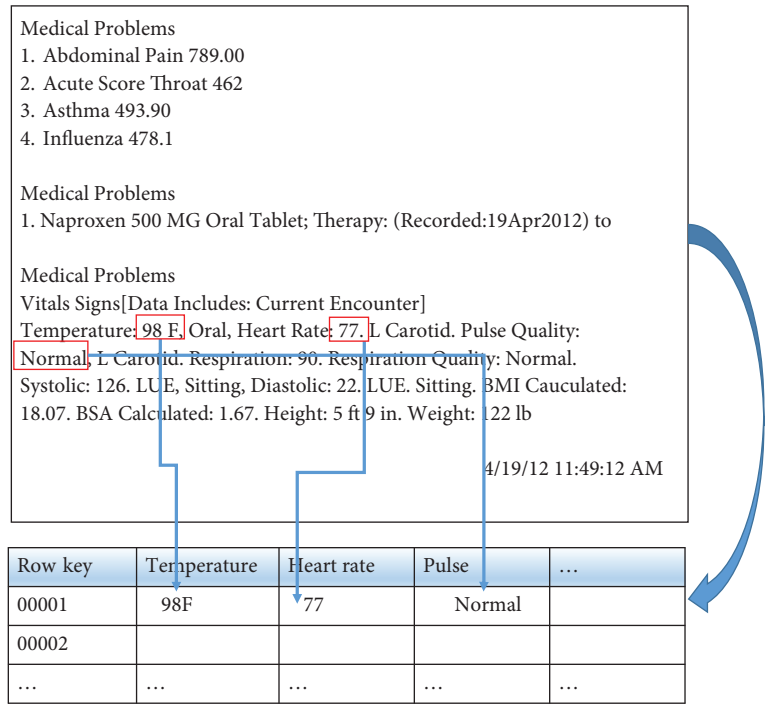


Figure 6: HBase example for unstructured data storage. Important information will be extracted from the raw data and stored in table format which is appropriate for further analysis.

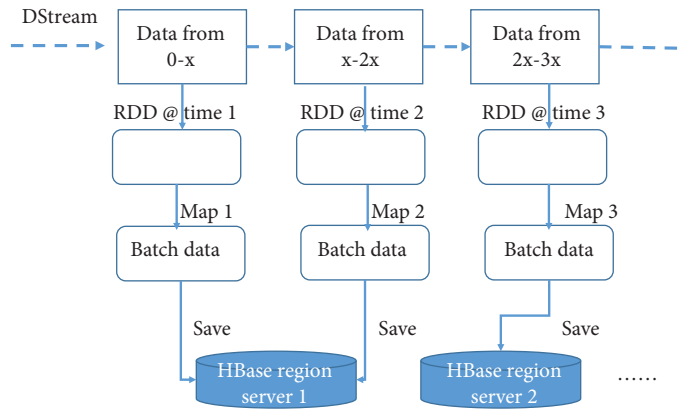


Figure 7: Streaming-data processing by Spark. The Spark engineer divides the stream data into sequential blocks and these blocks are stored in different nodes of the cluster.

4.2. Disease Data Statistical Analysis. Among the whole existing dataset, some of the patient data are treated as the training set for the disease-rule generation (some of the datasets are not disease related). The first step here is the counting of the diseases of the patients with their personal information, like their gender, age, nationality, and occupation. Since the data were stored in the HBase, the MapReduce framework was used to count the diseases. The training data

Condition attribute value | Decision attribute value

Row key	Temperature	Heart rate	Pulse	…	Disease
00001	98F	77L	Normal		Hypertension
00002	98.5F	80L	Normal		Diabetes I
00003	97F	70L	Fast		Diabetes II
00001	99.5F	79L	Normal		Bradycardia
…	…	…	…	…	…
00008					Hypertension
…	…	…	…	…	…

HBase region server 1
HBase region server 2
……

Figure 8: Training data example. Data comes from different sources and stored in different cluster nodes.

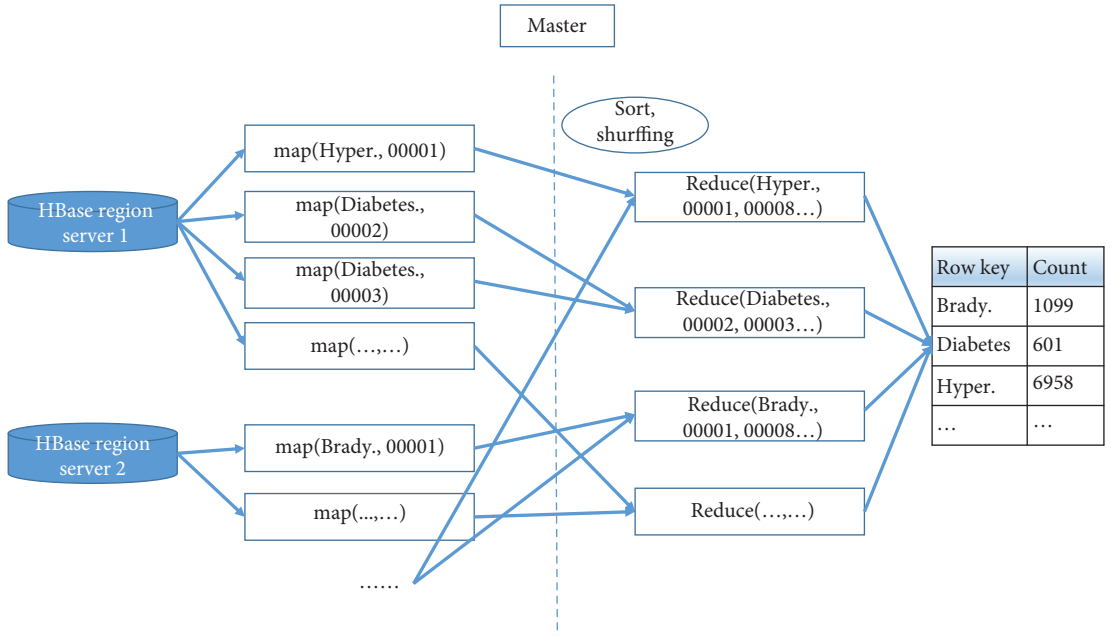


Figure 9: MapReduce procedure for disease count. Disease name is key and row ids are value for Map function. Disease name is key and all the row ids for the same disease are value for Reduce function. The count of row ids is the count for specified disease.

are stored in separate regional servers, as shown in Figure 8. The disease-count MapReduce procedure running in TPS is depicted in Figure 9.

The disease-count algorithm running in distributed environment is shown in Algorithm 1. It consists of two main procedures which are Map and Reduce. Map function is used to assign constant value 1 in terms of each row for distinct disease and patient. Reduce function is used to add all 1s together for the same disease, the sum of 1s is the count of the specified disease.

Based on the output, it is straightforward to obtain the patient list for a specified disease, as well as all of the personal information according to the patient ID.

4.3. Risk Factor Selection. The risk factor (RF) selection procedure is a process for feature selection. Hybrid feature selection has been applied to the raw dataset. First, we apply t-test

Input:
HBase table
Output:
Diseases count and related info
1. **class** Mapper
2. **method** map (HBase table)
3. **for** each instance row in table
4. **write** ((disease$_i$, patientID), 1)
5.
6. **class** Reducer
7. **method** reduce ((disease$_i$, patientID), ones[1,1,1,…n])
8. sum=0
9. **for each** one in ones do
10. sum+=1
11. **return** ((disease$_i$, patientID),sum)

Algorithm 1: Disease count running on HBase.

Input:
Hypertension disease data set
Output:
Selected features and SVM classifier
1. **load** data set
2. **Sample** the data randomly into training (67%) and testing(33%) data set
3. **set** target variable
4. generate the classifier based on the training data set
5. train the classifier using linear kernel function (similarity function)
6. predict the testing data set using the trained classifier
7. evaluate the classifier
8. Recursively select the features correspond to their weights

Algorithm 2: Linear SVM pseudocode.

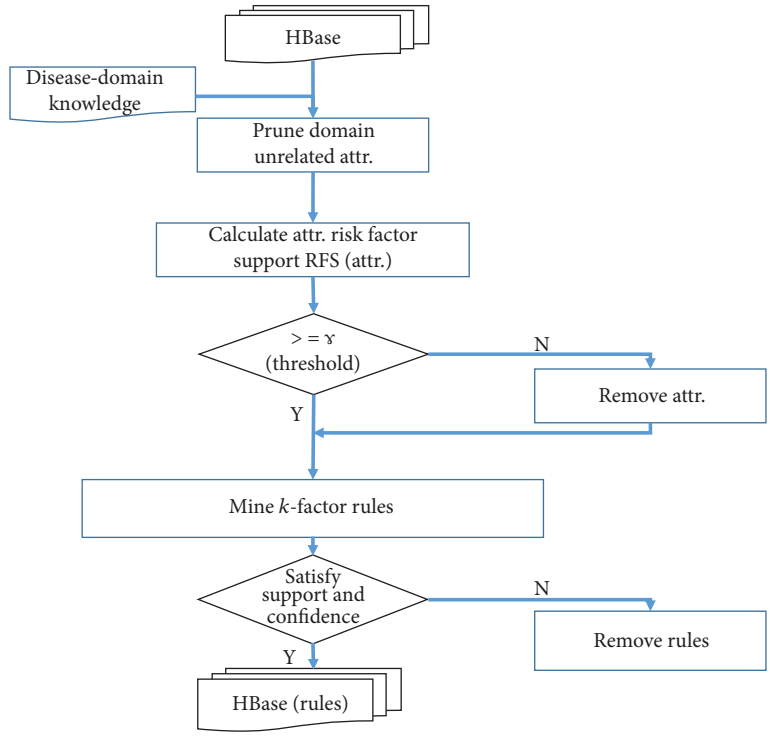


Figure 10: Disease-rule generation procedure. Domain expert's knowledge is used to prune the unrelated attributes for the first round. Then RFS is calculated for each remaining attributes to compare with predefined threshold γ, if larger than γ, it is treated as risk factor for further rule generation step.

combined with chi-squared test as filters to prune unrelated features according to research [11], chi-square and t-test are fast and can achieve relative high accuracy. The chi-squared test formula is,

$$\chi^2 = \sum_{i=1}^{k} \frac{(x_{i-} m_i)2}{m_i}, \tag{1}$$

where the expected numbers m_i were large enough known numbers in all cells assuming every x_i may be taken as normally distributed, and reached the result that, in the limit as n becoming large, χ^2 followed the chi-squared distribution with (k-1) degrees of freedom.

The t-test can be used, for example, to determine if two sets of data are significantly different from each other. The t-test

$$t = \frac{Z}{s} = \frac{(\bar{x} - \mu)/(\sigma/\sqrt{n})}{s}, \tag{2}$$

where $\bar{x}$ is the sample mean from a sample $X_1, X_2, \ldots, X_n$, of size n, s is the ratio of sample standard deviation over population standard deviation, σ is the population standard deviation of the data, and μ is the population mean.

After the first step, we propose a wrapper method based on linear support vector machines (SVMs). Firstly, we train the linear SVM on a subset of training data and retain only

Input:
Data partition D: a training set and associated class label C
Attribute list L (selected disease risk factors in previous step)
Output:
Decision tree with its root N
Method:
1. Create a node N,
2. **if** samples has the same class, C then,
3. **return** N as leaf node with class C label
4. **if** list of attributes is empty then
5. **return** N as leaf node with class label that is the most class in training set.
6. Choose test factor, that has the most GainRatio using attribute_selection_method
7. give node N with test-attribute label
8. **for each** attribute a_i in L
9. add branch in node N to test-attribute=a_i
10. make partition for sample s_i from training set where test-attribute=a_i
11. **if** s_i is empty then
12. attach leaf node with the most class in training set
13. **else** attach node that generated by Gnerate_decision _tree (s_i, L, test-attribute)
14. **return** N

Algorithm 3: C4.5 for important disease rule generation.

Table 2: Disease-count results.

Disease	Hyperten.	Dyslip.	Ostarthritis.	Diabetes	Asthm.
Count	9385	3213	5223	3500	923

Table 3: Statistical analysis.

	Weighted *N* (%)	Women (%)	Men (%)
Age			
20–44	20,131 (40.6)	11,416 (40.5)	8715 (41.5)
45–64	17,623 (35.6)	9946 (35.3)	7677 (36.6)
65+	11,775 (23.8)	6822 (24.2)	4593 (21.9)
Marital status			
Married	42,478 (86.6)	24,747 (88.5)	17,731(84.1)
Never married	6569 (13.4)	3208 (11.5)	3361 (15.9)
Education			
University	12,990 (28.9)	6612 (25.5)	6378 (33.6)
High school	14,876 (33.1)	8129 (31.3)	6747 (35.6)
Middle school	4950 (11.0)	2659 (10.2)	2291 (12.1)
Elementary school	12,125 (27.0)	8568 (33.0)	3557 (18.8)
Occupation			
Office worker	8993 (20.1)	4104 (15.8)	4889 (25.9)
Manual worker	17,181 (38.4)	8099 (31.3)	9082 (48.2)
Unemployed	18,602 (41.5)	13,710 (52.9)	4892 (25.9)
Income Level			
1st quartile	12,077 (24.8)	6862 (24.8)	5215 (24.9)
2nd quartile	12,183 (25.1)	6940 (25.1)	5243 (25.0)
3rd quartile	12,193 (25.1)	6930 (25.1)	5263 (25.1)
4th quartile	12,167 (25.0)	6912 (25.0)	5255 (25.1)
Hypertension			
No	8619 (47.9)	4954 (47.4)	3665 (48.5)
Yes	9383 (52.1)	5488 (52.6)	3895 (51.5)

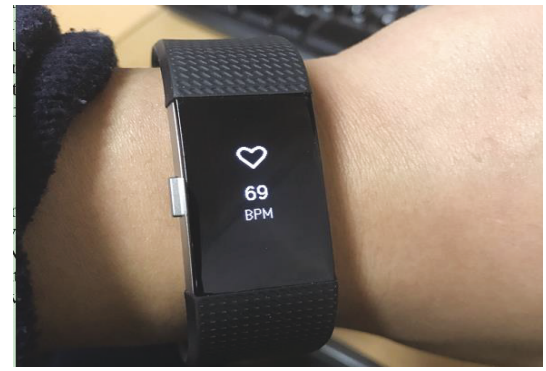


Figure 11: Screenshot of the IoT device we used in our system.

those features that correspond to highly weighted components (in absolute value sense) of the normal to the resulting hyperplane that separates positive and negative examples for the class. Secondly, recursively eliminate features whose weight value is close to zero. Finally, the features remained are selected as risk factor candidates. We created a representation result in experimental session, Chapter 5. The pseudocode is given in Algorithm 2 below. First, we divide the original dataset into training and testing dataset; the SVM classifier is generated based on this training dataset. After evaluating the classifier, the features will be recursively selected according to the weight until the stop criterion has been met.

The result will be described in experimental session.

4.4. Disease-Rule Generation. The format of disease rules is like that of the IF THEN rule; for example, IF (edu = elementary, B1 < =0.86 mg/day, married), THEN (hypertension = yes). The purpose here is the mining of all of the disease-related rules from the training dataset for a further data analysis including disease prediction and disease

region	town_t	apt_t	sex	age	incm	ho_incm	edu	occp	kstrata	cfam	genertn	allownc	house	live_t	ainc_unit1
ainc	marri_1	marri_2	fam_rela	tins	npins	M_1_yr	mt_nontrt	BH9_11	BH9_13	BH1_1	BH1_2	BH1_3			
BH1_8	BH1_6	BH2_61	BH2_62	BH2_63	BH2_66	BH2_67	BH2_64	LQ4_01	LQ4_02	LQ4_03					
LQ4_04	LQ4_05	LQ4_06	LQ4_07	LQ4_08	LQ4_09	LQ4_10	LQ4_11	LQ4_12	LQ4_13	LQ4_14					
LQ4_15	LQ4_16	LQ4_21	LQ4_22	LQ4_23	LQ4_17	LQ4_18	LQ4_19	LQ4_20	LQ1_sb	LQ2_ab					
LQ2_mn	LQ_1EQL	LQ_2EQL	LQ_3EQL	LQ_4EQL	LQ_5EQL	EQ5D	graduat	EC1_1	EC1_2	EC_occp	EC_stt_1	EC_stt_2			
EC_wh	EC_wht_0	EC_wht_23	EC_wht_5	EC_lgw_2	EC_lgw_4	EC_lgw_5	EC_pedu_1	EC_pedu_2	BO1	BO1_3	BO2_1				
BO3_01	BO3_02	BO3_03	BO3_14	BO3_05	BO3_04	BO3_12	BO3_07	BO3_09	BO3_10	BD2	BD1_11				
BD2_1	BD2_31	BD2_32	BD7_4	BD7_6	BD7_5	dr_month	BA2_12	BA2_13	BA1_3	BA1_5					
BA2_2_1	BA2_2_2	BA2_2_5	BA2_2_6	BA2_22	sc_seatblt2	BP8	BP1	BP7	mh_stress	BS1_1	BS3_1	BS3_3	BS6_2	BS6_2_1	
BS6_2_2	BS6_3	BS6_4	BS6_4_1	BS6_4_2	BS5_2	BS5_21	BS5_22	BS5_24	BS5_26	BS5_28	BS5_32	BS5_25			
BS5_27	BS5_29	BS5_30	BS8_2	BS9_1	BS9_2	BS13	BS12_1	BS12_2	BE3_71	BE3_73	BE3_74	BE3_81			
BE3_82	BE3_83	BE3_84	BE3_91	BE3_92	BE3_93	BE3_94	BE3_75	BE3_76	BE3_77	BE3_78					
BE3_85	BE3_86	BE3_87	BE3_88	BE8_1	BE8_2	BE3_31	BE3_32	BE3_33	BE5_1	BE5_2	pa_aerobic	LW_ms			
LW_mp_a	LW_ms_a	LW_pr	LW_pr_1	LW_mt	LW_mt_a1	LW_mt_a2	LW_br	LW_br_ch	LW_br_dur	LW_br_yy					
LW_br_mm	HE_HP	O_DMFTP	O_DMFIP	OR1	O_ortho	BM1_0	BM1_1	BM1_2	BM1_3	BM1_4	BM1_5				
BM1_6	BM1_7	BM1_8	BM2_3	BM2_2	BM2_4	BM2_5	BM13	BM7	O_chew_d	BM8	OR1_2	MO4_00			
BM12	BM12_1	T_Q_SNSTDG	T_Q_SNSTPT	T_Q_OMPT	T_Q_HR	T_Q_HR1	T_Q_VN	T_Q_VN1	T_NQ_PH						
T_NQ_PH_T	T_NQ_OCP	T_NQ_OCP_T	T_NQ_OCP_P	T_NQ_LS	T_NQ_LS_T	T_NQ_FIR	T_NQ_FIR_P	GS_use	GS_mea_r_1						
TH_ult	TH_ult_1	TH_ult_2	TH_deli_1	L_BR	L_DN	L_BR_FQ	L_DN_FQ	L_OUT_FQ	L_BR_TO	L_BR_WHO	L_LN_TO				
L_LN_WHO	L_DN_TO	L_DN_WHO	LK_LB_CO	LK_LB_US	LK_LB_IT	LK_LB_EF	LF_CARE	N_DIET	N_DIET_WHY						
N_DUSUAL	N_PRG	N_INTK	N_EN	N_WATER	N_PROT	N_FAT	N_SFA	N_MUFA	N_PUFA	N_N3	N_N6	N_chol			
N_tdf	N_CA	N_PHOS	N_NA	N_K	N_VA	N_RETIN	N_B1	N_B2	N_NIAC	LF_CHIL	LF_SAFE	LF_S2	LF_S3	LF_S4	LF_S5	LF_S5_1
LF_S6	LF_S7	LF_S8	LF_S9	LF_S10	LF_S11	LF_S12	LF_S13	LF_S14	LF_S15	LF_S16	LF_SECUR	LF_SECUR_G			

Figure 12: Features selected after filters methods: *t*-test and chi-squared test. This result is used as the input for wrapper method.

Table 4: SVM-based attribute risk factor computation.

Number	Ranked risk factors	Weight
1	Age	0.0890
2	DE1_pr: current status of diabetes	0.0885
3	DN1_dg: kidney failure diagnosis	0.0881
4	BD1_11: (12 years old or older) frequency of drinking for 1 year	0.0873
5	BP8: average sleep time per day	0.0841
6	BP1: usually stressed	0.0370
7	BS3_1: (adult) currently smoking	0.0322
8	N_NA: sodium intake (mg)	0.0321

Table 5: Hypertension-disease rules generated by C4.5.

Number	Rules	Conf.
1	If (age > 70, DE1_pr = 1) - >yes	0.86
2	If (46.5 < age < = 70, BD1_11>=6) - >yes	0.74
3	If (BP8 >8, N_B1 < = 0.88 mg) - >yes	0.73
4	If (marri_1=2, BD1_11>=6) - >yes	0.73
5	If (age <= 46.5, BD1_11<4, N_WATER > 1060.5 ml/day) - >no	0.72
6	If (DN1_dg=1, marri_1=2) - >yes	0.63
7	If (N_B1 < = 0.86 mg/day, N_WATER< = 485.63) - >yes	0.61
8	If (marri_1=1, N_NA >= 3532.6, N_B1 < = 0.86) - >yes	0.60

detection. Another concept that needs to be described is the *k*-factor rule, where *k* is the number of risk factors. The rule here is a three-factor rule.

Based on the key RF, the procedure for the disease-rule generation is shown in Figure 10.

Combined with the disease-domain knowledge that is provided by the domain experts, the TPS has the power to ignore a large amount of the attributes at the very beginning, thereby leaving only the high risk factor support (RFS) attributes [15]. Among these attributes, two attribute sets are

formed for a comparison based on the correlation; that is, if they are very similar to each other, they are strongly correlated, and the TPS will remove the one with the low RFS. Then, a basic association rule-mining algorithm like Apriori and commonly used decision-tree algorithms like C4.5, CART, or Random Forest will be used to generate the k-RF rules. For the first stage of the research, however, only C4.5 is used for the testing data. The algorithmic pseudocode is given in Algorithm 3. First, we calculate the GainRatio for each risk factor r in L, then we choose the factor with the highest GainRatio and create a decision node based on this factor, split the dataset by this node. We repeat these steps until all nodes with appropriate satisfied GainRatio value have been used to generate the tree. The path from the root to leaf is the disease rule.

The reasons for the selection of C4.5 are its simplicity, the accuracy of its results, and the ability to use it on numerical and categorical attributes even with the presence of an overfitting problem.

During the next research stage, algorithms including CART, Random Forest, and KNN will be tested to find the one that fits the most datasets with a high accuracy. Finally, the generated rules will be stored in the HBase in preparation for the next few steps.

4.5. Disease Prediction and Detection. According to the disease rules and the RFS, the highly related key RFS, such as heavy drinking for hypertension, are used to generate the prediction model. This work was completed in the authors' previous study [15]. The multi-RFS will be compared with the disease rule to confirm the patient health condition, and again, this work was completed in [15].

4.6. Cloud-Service Module. The Cloud module plays the roles of data storage and transfer and consists of the public Cloud services GCSql and FCM. Considering the efficiency, connection, and security problems, only commonly used, important, and urgent information like an urgent message from the health-care provider is stored in the GCSql database. Alternatively, the FCM is salient for the communication between the TPS and any relevant device. Again, this work was completed in [15].

5. Experiment

At this stage, the detailed design work of the whole system has been finished, while the implementation work is partially finished. Further, the mobile health-sensor network has been set up in the experimental environment. A large amount of simulated data and a small number of real data that were downloaded from the Korea National Health and Nutrient Examination Survey (KNHANES) [24] were combined with simulated dataset, and the testing data comprises approximately 60,900 patient records, including basic personal information, disease information, and clinical information. The entire cluster has been established, and it can interact with Android devices through the Cloud module. Several devices have been used for the purpose of testing. Meanwhile, an app has been developed for the data collection and a dataset statistical-analysis-result visualization.

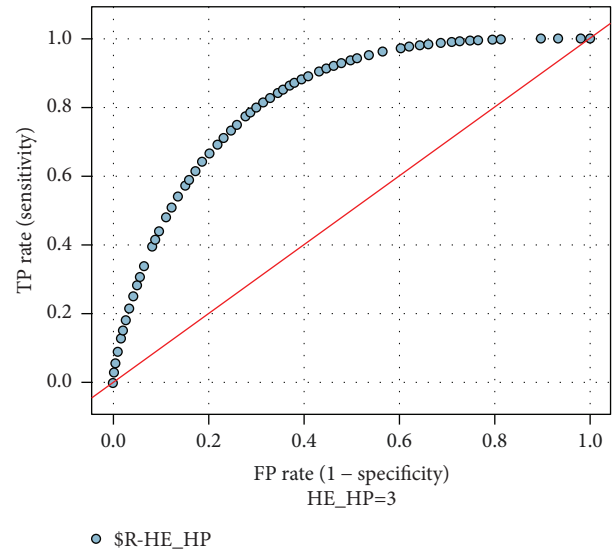


Figure 13: AUC of the trained SVM model. HE_HP = 3.0 is the target attribute.

Table 6: Performance comparison of the two algorithms.

Method	Number of attr.	Sensitivity	Specificity	Accuracy
C4.5	81	0.448	0.692	0.688
SVM	80	0.451	0.705	0.698

A MapReduce-based algorithm called "Disease Count" has been implemented and the pseudocode is given in Algorithm 1. The result of the Disease Count algorithm is given in Table 2. The hypertension data that contains 9383 records has been used as the test dataset, and its statistical-analysis results are given in Table 3. The number of main attributes is 3, but not all of the results are listed in this table due to a space limitation.

The advanced IoT device Fitbit Charge 2 has been used for user sports, sleeping, pulse, and breath detection. The model we used is shown in Figure 11.

Hybrid feature selection mechanism has been used to select highly related features. Among all these attributes, t-test and chi-squared test implemented by using R language have been applied to select the key RFS for hypertension. Finally, 217 features have been selected among all 526 features. The results are given in Figure 12.

SVM-based wrapper feature selection method has been applied to the attributes selected from the previous step. Therefore, 81 features have been selected and the top 8 features are given in Table 4.

For the disease rule generation procedure, the minimum support threshold was set to 0.1, and the minimum confidence threshold was set to 0.3. C4.5 was used to generate the hypertension-disease rules that are shown in Table 5. DI1_dg = 0 means hypertension not diagnosed, while DI1_dg = 1 means hypertension diagnosed. The result is given in Table 5.

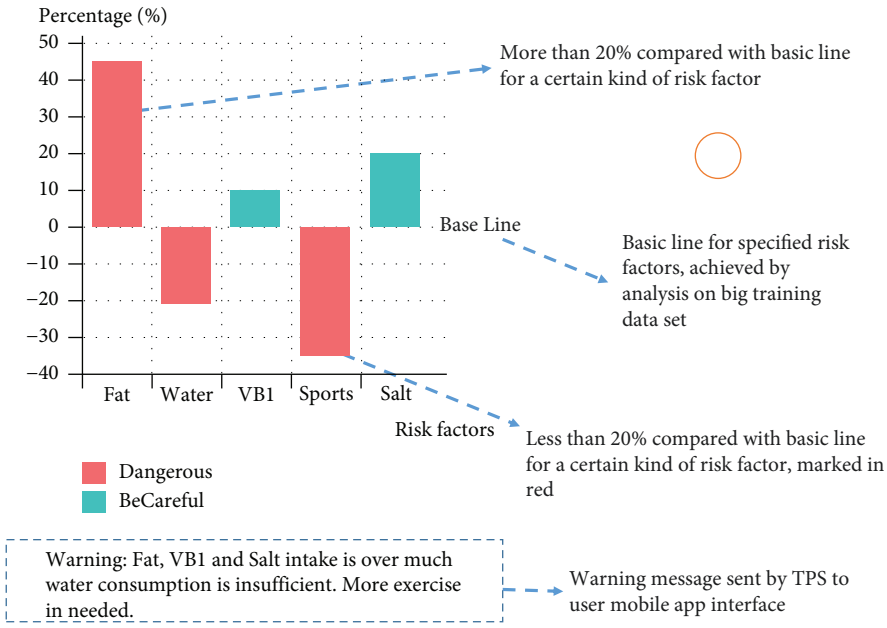


Figure 14: Rule with the risk factor (RF) analysis result visualization. The RFS marked in red mean too much or too few compared with the standard.

For the purpose of the comparison, SVM classification algorithm is also applied to predict the hypertension. We trained one model considered for the variable "HE_HP" for regressor following all the other 80 variables. The support vector machine results were implemented using the "e1071" package in R by performing a 10-folder cross validation using radial basis kernel function which the best parameters are: cost = 1000, gamma = 0.01. The result is given in Figure 13.

We have compared the accuracy of the two methods in terms of sensitivity, specificity, and accuracy. The result is shown in Table 6 below,

From the results, it is possible to draw several conclusions as follows: (1) age and alcohol intake play very important role for hypertension; there is great chance for the elder and heavy drinkers having this disease. (2) The effect of smoking for hypertension is inferior to alcohol. (3) The elders should have light food instead of salty food. (4) SVM performs better result C4.5 in our dataset.

Nevertheless, the accuracy of both algorithms is not satisfactory. This is due to the challenges of collecting big healthcare data, and this issue has been depicted in introduction session. For the next stage of research, we will focus on solving this problem.

The analysis results are directly and visually displayed in the user devices. The interpretations of the analysis results are shown in Figure 14. The authors have published another paper [15] wherein a simple disease-rule visualization method is discussed, since it is also a challenging work.

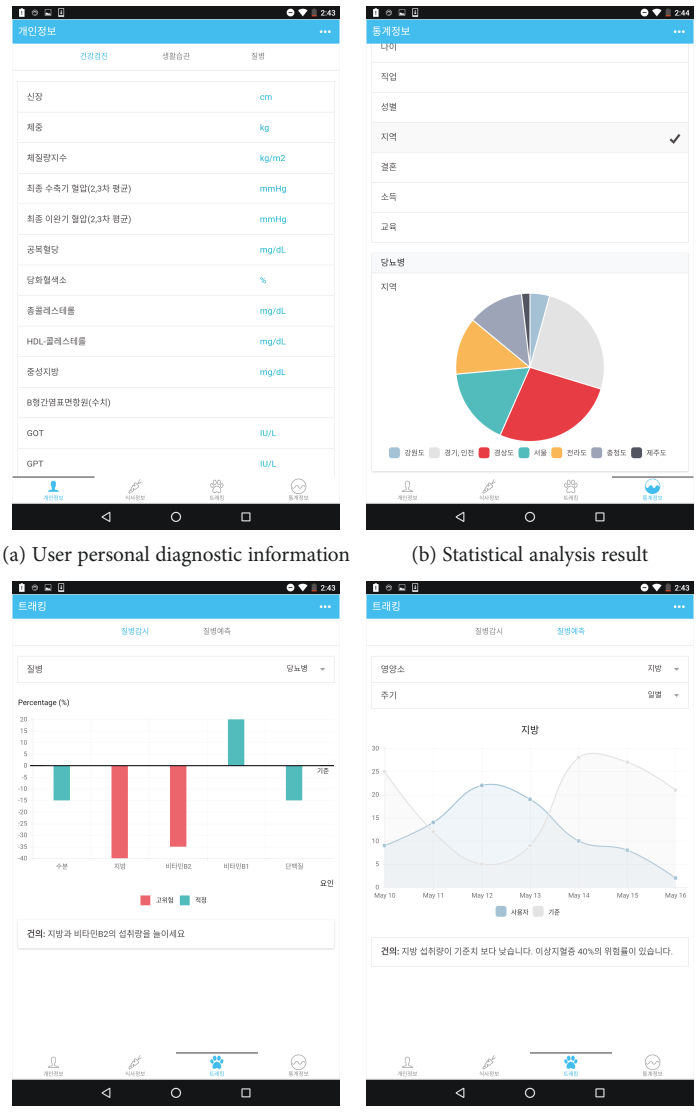
Figure 14 illustrates the disease-detection-result visualization interface of the designed app of this study. The x-axis of the coordinate lists rank the key RFS of a certain kind of disease, and it is also the basic line that is based on the training big data analysis result (e.g., a standard factor like nutritional intake will be visible for a healthy patient, but the concrete value will be hidden from the figure). The y-axis is the percentage of the intake that exceeds or is inferior to the standard factor intake; the disease rules consist of these factors. For a certain disease, there is usually more than one rule (consisting of RFS) that is related to the disease. Compared with these rules, if the matching rate $> \beta$ (expert-defined threshold, e.g., 80%), the system will treat this patient segment as the disease holder.

The figures below are GUI of our system, which is implemented based on Ionic using hybrid programming techniques. The characters are in Korean since it is mainly developed for Korean users so far. Selected user interfaces are given in Figure 15.

6. Conclusion and Future Work

In the present work, Hadoop, Spark, and DM techniques are exploited to design a comprehensive, real-time, and intelligent mobile health-care system that can facilitate a step-by-step process for disease detection and prediction. The purpose of this work is the provision of a practical assistant system for self-based user health care, as well as the design of a complementary system for patient disease diagnosis. During the experiment section, firstly, disease data stored in distributed environment has been retrieved by MapReduce method and analysed by statistical method to give us an overview of the data. Then both statistical methods and DM methods are used to select the features related to hypertension disease. These attributes are the risk factors as well. Based on these factors, C4.5 and SVM methods are used to generate the classifier model for disease prediction. Finally, we displayed the analysis result on users' mobile devices.

An overview and a guide are described in detail for a future work as well. For the next stage of research, after the implementation of the whole system, in-depth simulations will be performed to validate the systemic performance in



(a) User personal diagnostic information

(b) Statistical analysis result

(c) Disease detection interface

(d) Disease prediction interface

Figure 15: User interfaces of the system.

terms of the application of the proposed system in a real environment. The TPS and algorithms like C5, Random Forest, and other algorithms will be run on the TPS cluster to compare the efficiency and the accuracy. The procedure for the disease detection and prediction will be optimized continuously and extended to other chronic diseases. Finally, it is hoped that this system will contribute to health-care academia as well as the industry.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant no.2017R1A2B4010826) and the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2013-0-00881), supervised by the IITP (Institute for Information & Communication Technology Promotion) and also supported by the National Natural Science Foundation of China (61702324).

References

[1] A. V. Chobanian, G. L. Bakris, H. R. Black et al., "Seventh report of the joint national committee on prevention, detection, evaluation, and treatment of high blood pressure," *Hypertension*, vol. 42, no. 6, pp. 1206–1252, 2003.

[2] P. D. Kaur and I. Chana, "Cloud based intelligent system for delivering health care as a service," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 346–359, 2014.

[3] H. Horiguchi, H. Yasunaga, H. Hashimoto, and K. Ohe, "A user-friendly tool to transform large scale administrative data into wide table format using a MapReduce program with a

pig Latin based script," *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, p. 151, 2012.

[4] R. C. Taylor, "An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics," *BMC Bioinformatics*, vol. 11, article S1, Supplement 12, 2010.

[5] M. C. Schatz, "CloudBurst: highly sensitive read mapping with MapReduce," *Bioinformatics*, vol. 25, no. 11, pp. 1363–1369, 2009.

[6] V. Sharma, K. Mankodiya, F. De La Torre et al., "SPARK: personalized Parkinson disease interventions through synergy between a smartphone and a smartwatch," in *Design, User Experience, and Usability. User Experience Design for Everyday Life Applications and Services. DUXU 2014*, A. Marcus, Ed., vol. 8519 of Lecture Notes in Computer Science, pp. 103–114, Springer, Cham, 2014.

[7] H. Kim, M. I. M. Ishag, M. Piao, T. Kwon, and K. H. Ryu, "A data mining approach for cardiovascular disease diagnosis using heart rate variability and images of carotid arteries," *Symmetry*, vol. 8, no. 6, p. 47, 2016.

[8] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "RFID technology for IoT-based personal healthcare in smart spaces," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 144–152, 2014.

[9] M. J. Huang, M. Y. Chen, and S. C. Lee, "Integrating data mining with case-based reasoning for chronic diseases prognosis and diagnosis," *Expert Systems with Applications*, vol. 32, no. 3, pp. 856–867, 2007.

[10] S. H. Ha and S. H. Joo, "A hybrid data mining method for the medical classification of chest pain," *International Journal of Computer and Information Engineering*, vol. 4, no. 1, pp. 33–38, 2010.

[11] H. J. Aboumatar, K. A. Carson, M. C. Beach, D. L. Roter, and L. A. Cooper, "The impact of health literacy on desire for participation in healthcare, medical visit communication, and patient reported outcomes among patients with hypertension," *Journal of General Internal Medicine*, vol. 28, no. 11, pp. 1469–1476, 2013.

[12] Y. Piao, M. Piao, C. H. Jin et al., "A new ensemble method with feature space partitioning for high-dimensional data classification," *Mathematical Problems in Engineering*, vol. 2015, Article ID 590678, 12 pages, 2015.

[13] M. Herland, T. M. Khoshgoftaar, and R. Wald, "Survey of clinical data mining applications on big data in health informatics," in *2013 12th International Conference on Machine Learning and Applications*, vol. 2, pp. 465–472, Miami, FL, USA, December 2013.

[14] R. Fang, S. Pouyanfar, Y. Yang, S. C. Chen, and S. S. Iyengar, "Computational health informatics in the big data age: a survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, article 12, 2016.

[15] D. Li, H. Park, E. Batbaatar, Y. Piao, and K. Ryu, "Design and partial implementation of health care system for disease detection and behavior analysis by using DM techniques," in *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pp. 781–786, Auckland, New Zealand, August 2016.

[16] "Welcome to Apache™ Hadoop," http://hadoop.apache.org/.

[17] "Apache™ Spark, a fast and general engine for large-scale data processing," http://spark.apache.org/.

[18] J. Han, J. Pei, and M. Kamber, *Chapter 1 Introduction Data Mining: Concepts and Techniques*, Elsevier, Waltham, MA, USA, 2011.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] H. H. Hsu, C. W. Hsieh, and M. D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144–8150, 2011.

[21] "Apache Sqoop™," http://sqoop.apache.org/.

[22] B. Lee and E. Jeong, "A design of a patient-customized healthcare system based on the hadoop with text mining (PHSHT) for an efficient disease management and prediction," *International Journal of Software Engineering and Its Applications*, vol. 8, no. 8, pp. 131–150, 2014.

[23] "Apache Spark™," http://spark.apache.org/streaming/.

[24] "KNHANES," https://knhanes.cdc.go.kr/knhanes/sub03/sub03_02_02.do.