symmetry

MDPI

Article

System Framework for Cardiovascular Disease Prediction Based on Big Data Technology

Sang Hun Han [1], Kyoung Ok Kim [2], Eun Jong Cha [3], Kyung Ah Kim [3,*] and Ho Sun Shon [4,*]

[1] Department of Computer Science, College of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, Korea; likelamb@gmail.com
[2] Department of Nursing, Woosong College, Daejeon 34518, Korea; kokim@wsi.ac.kr
[3] Department of Biomedical Engineering, College of Medicine, Chungbuk National University, Cheongju 28644, Korea; ejcha@chungbuk.ac.kr
[4] Medical Research Institute, College of Medicine, Chungbuk National University, Cheongju 28644, Korea
* Correspondence: kimka@chungbuk.ac.kr (K.A.K.); shon0621@gmail.com (H.S.S.);
Tel.: +82-43-261-2852 (K.A.K.); +82-43-261-2841 (H.S.S.)

Received: 25 October 2017; Accepted: 24 November 2017; Published: 27 November 2017

Abstract: Amid growing concern over the changing climate, environment, and health care, the interconnectivity between cardiovascular diseases, coupled with rapid industrialization, and a variety of environmental factors, has been the focus of recent research. It is necessary to research risk factor extraction techniques that consider individual external factors and predict diseases and conditions. Therefore, we designed a framework to collect and store various domains of data on the causes of cardiovascular disease, and constructed a big data integrated database. A variety of open source databases were integrated and migrated onto distributed storage devices. The integrated database was composed of clinical data on cardiovascular diseases, national health and nutrition examination surveys, statistical geographic information, population and housing censuses, meteorological administration data, and Health Insurance Review and Assessment Service data. The framework was composed of data, speed, analysis, and service layers, all stored on distributed storage devices. Finally, we proposed a framework for a cardiovascular disease prediction system based on lambda architecture to solve the problems associated with the real-time analyses of big data. This system can be used to help predict and diagnose illnesses, such as cardiovascular diseases.

Keywords: big data; cardiovascular disease; integrated database; prediction system

1. Introduction

As climate conditions change and concern over health care grows, health care-related issues such as cardiovascular diseases have exhibited close correlations with external environmental factors. Globally, the prevalence and risk of cardiovascular disease has increased markedly, accompanied with rapid industrialization, and is now a major problem of aging societies. Several studies have identified a relationship between fine dust ($PM_{2.5}$) exposure and cardiac infarction based on correlations between environmental pollution and cardiovascular disease [1–3], as well as increases in the hospitalization rates of patients with cardiovascular disorders according to temperature and carbon dioxide concentration [4]. Moreover, environmental pollution by nitrogen dioxide and sulfur dioxide increases mortality [5]. Accordingly, it is necessary to create efficient disease prediction and risk estimation techniques to prevent cardiovascular diseases, which are correlated with a variety of external environmental factors.

The existing cardiovascular disease prediction system considers lifestyle factors such as smoking, drinking, diet, exercise, and stress [6]; however, according to recent research, many additional factors affect cardiovascular disease such as climate change, health conditions, social and economic factors, and atmospheric environment information. Yet, no integration database (DB) based on diverse environmental factors can be used to extract risk factors that occur in different areas. Moreover, no optimal prediction system exists that considers such a variety of factors.

Because recent research has identified the environmental factors that harm health, such as abnormal climate and atmospheric pollution, it is necessary to integrate this information into quantitative evaluations and diagnoses of health, to provide a systematic health care policy. In particular, it is increasingly important for health and environmental policies to consider these factors to minimize risks. In 2013, the UK Department of Health announced the Personalized Health and Care 2020 framework, which reinforced control over medical treatment and welfare information for patients, and constructed the Health & Social Care Information Center as an independent organization that collects, stores, connects, and analyzes distributed social security data [7]. As another example, Roski et al. used a personalized medical care clinical decision support system and anticipated service optimization that reflected patient data. This enabled the practical application of health big data for population health analyses and prevention [8].

Numerous studies have examined the correlations between health conditions and cardiovascular disorders, researching cardiac disorder diagnosis and prediction systems by using artificial neural networks, data mining [9,10], and the association rule (i.e., emerging patterns) to identify significant patterns among medical treatment data [11]. Furthermore, a variety of studies have examined integrated DBs of large volumes of data, approaching the method from an ontology perspective [12] using open source-based research [13].

Studies have also evaluated changes in environmental pollutant concentrations, and the degree of health damage according to national health insurance data and climate changes; examined the use of danger and risk maps in environment health regions in environmental offices; and evaluated frameworks for processing big data appropriate to each domain [14]. It has been found that such data can be implemented by using the MapReduce model in Hadoop, yielding excellent scalability [15,16].

Lambda architecture is a big data technique that can be used to support real-time analyses; however, it has the limitation of not being able to analyze a large volume of data in real time. To address this limitation, a method can be employed that blends data made in advance in a batch layer with data processed in real time. Then, the data can be generated and stored. To achieve this, data are formed in batch view in a cycle with a batch layer, and identical data are formed in real-time view via real-time data processing. These two data sets are then blended and analyzed, enabling the analysis of data that reflects real-time data [17,18]. Supporting this, Amazon Web Services, which processes big data, wrote a White Paper on the integration of batch processing and real-time processing into a single network using lambda architecture [19].

To understand big data and streaming data analysis, the Apache Hadoop software library and Microsoft Azure provide a variety of solutions and comprehensive analysis techniques [20,21]. Moreover, real-time analyses have been performed by using key value analysis [22], streaming analysis using Apache Spark, and network analysis using the Open Network Operating System controller in real time [23,24]. YARN, which is used in batch processing, is a resource management platform of Hadoop that influences the energy efficiency of a cluster and utility of the application [25,26]. Furthermore, NoSQL (Not Only SQL database) of Hadoop can be used in the service layer, and a new indexing generation and storage technique for the Hadoop echo system has been developed that achieves better performance in the Hadoop environment [27–29]. In addition, performance development modeling has been carried out using multiple solution techniques based on lambda architecture and models managing social and crowding emergencies [30,31].

Currently, the collection and storage of large amounts of heterogeneous data from different domains involves difficulties not easily processed in simple DBs; therefore, in this study, we aimed to create an effective processing and analytical technique based on big data. We did not extract the risk factors of clinical data, but rather designed a prototype disease prediction system driven by a complex set of factors. This system includes risk factors from a variety of domains, including environmental, health, clinic, population, and climate conditions. Then, we integrated the DB of these various domains and developed a prototype prediction system based on complex factors from patients with cardiovascular disorders. The specific aim was to solve these problems based on lambda architecture. Specifically, the contributions of this study are threefold: (1) it collects data on various factors that can affect cardiovascular disease; (2) it designs and implements integrated DBs based on big data; and (3) it offers a prototype design of an analysis system that can predict cardiovascular disease.

2. Methods

Figure 1 presents an overview of the proposed prediction system. The data layer is composed of a data integration engine that enables the migration of a variety of data into distributed storage devices. The data integration engine synthetically stores various properties and data with diverse structures, and then performs preprocessing to make the data suitable for analysis and reformation in the analysis layer. The speed layer is composed of a real-time integration engine that preprocesses data generated in real time, and delivers the results analyzed through a real-time analysis model to the analysis layer. The analysis layer is composed of a data analysis engine, which analyzes the data collected in the data layer with an analysis model pipeline, and then merges the analysis results with those from the speed layer. It then uses these results as the input value for the prediction model pipeline, and delivers the analysis results to the service layer through the prediction model pipeline. The service layer provides users with a hybrid web or hybrid mobile app by using the analysis results in the analysis layer. All the data in each layer are stored on distributed environmental storage devices.

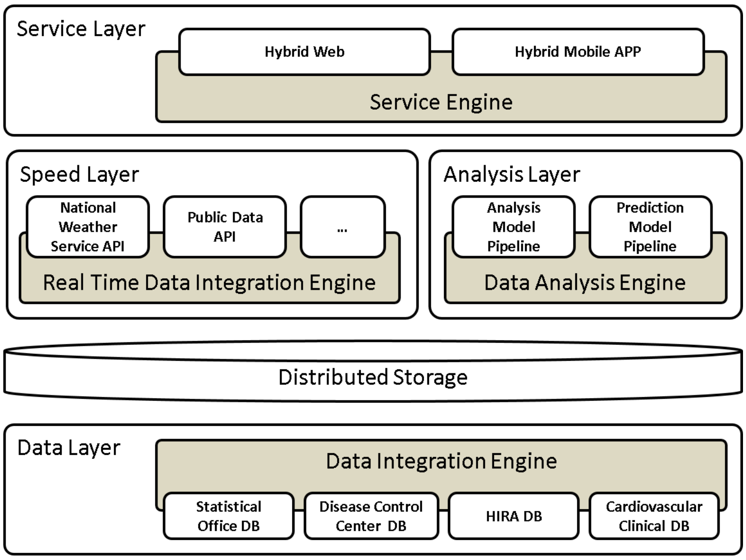


Figure 1. Overview of the proposed prediction system.

2.1. Data Integration Engine

The data layer is composed of the data integration engine, which migrates DB or file data such as the national health and nutrition survey provided by the Korea Centers for Disease Control and Prevention and data from the HIRA (Health Insurance Review and Assessment Service) onto distributed storage devices. The data integration engine is implemented into a "map-side only job" by using the MapReduce framework of Hadoop; it uses the computing power of the nodes comprising the Hadoop cluster in parallel.

Algorithm 1 presents the map-side only job implemented in the data integration engine. It accepts various data IDs and records them as input values, after which they are printed out and distributed to storage devices, after performing preprocessing and integration according to the ID of records in each record column. Figure 2 shows the data integration engine in which each DB is composed of optimized map-side only jobs. Each map-side only job extracts the data of a column composed of the records of each DB, and these are stored on distributed storage devices with the ID of each DB after preprocessing, such as normalization.

Algorithm 1: Preprocessing and Integration

```
class MAPPER
    method MAP (dbid $id$, record $R$)
        $k \leftarrow R$.ID
        $r \leftarrow R$.RECORD
        $r' \leftarrow$ record (null)
        for all column $c \in$ record $r$ do
            column $c' = $ Preprocessing ($k$, $c$)
            column $c'' = $ Integration ($k$, column $c'$)
            $r' = $ pair ($r'$, $c'$)
      EMIT (pair($k$, $c''$), $r'$)
```

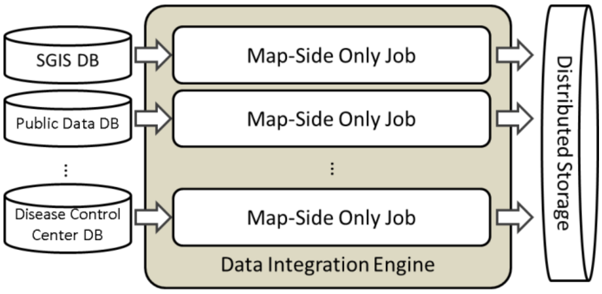


Figure 2. Data integration engine.

2.2. Real-Time Data Integration Engine

The real-time data integration engine composed of the speed layer provides and collects data in real time by using an API (Application Program Interface), such as the SGIS (Statistical Geographic Information Service) API of Statistics Korea, which provides data in the form of a query, or public data API. The collected data constitute the streaming data analysis pipeline based on Apache Spark. As shown in Figure 3, the real-time data integration engine collects data in real time, and verifies redundancies with data already collected and analyzed. Then, to ensure that only newly generated data are sent with the Spark streaming job to each DB, it is composed of an API manager, data distributor, and each Spark streaming job.

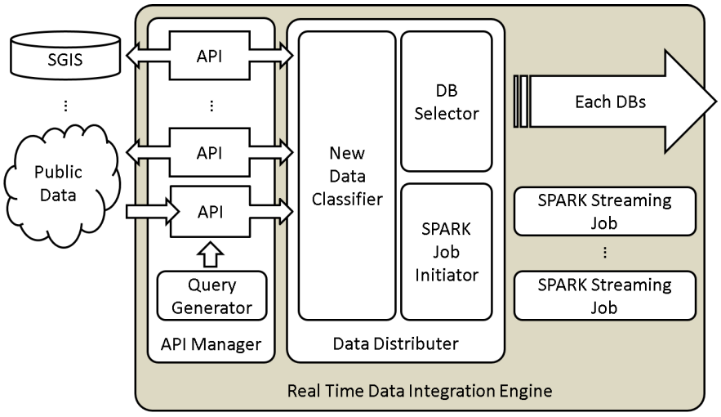


Figure 3. Real-time data integration engine.

2.2.1. API Manager

The API manager in the first data streaming step of the real-time data integration engine is composed of APIs optimized to source all real-time data. Each API is implemented according to the requirements of the organization providing the API, and some APIs provide data based on the query. To do this, the query generator generates a query according to the API. In addition, the query generator can dynamically generate a query according to user requirements in the service layer.

2.2.2. Data Distributer

The data distributer classifies the data collected from each API in the API manager and delivers data into the data integration engine and Spark streaming engine. The data collected through each API can be delivered with redundant data already collected according to each API condition. This is achieved with the data classifier of the data distributer. The data distributer modifies the data collected through the data classifier or classifies new data, while the Spark job initiator executes the Spark streaming job for analysis and the DB selector selects the suitable DB data and stores them in the DB.

2.3. Data Analysis Engine

The data analysis engine involving the analysis layer is composed of an optimized analysis pipeline that identifies correlations between factors and cardiovascular disorders, while the pipeline for the prediction model is based on the analysis results. As shown in Figure 4, the data analysis engine is composed of the analytics pipeline, which optimizes the analytics pipeline and prediction model with optimized analysis models, and the prediction pipeline that generates a knowledge DB for prediction recommendations.

The analytics pipeline, which is composed of a variety of models optimized for analysis, sends each analysis result to the Score Manager. The Score Manager then delivers the analysis results to the feedback generator, which manages the analysis results and analysis model improvement, and the prediction model optimizer, which further optimizes the prediction model. The prediction pipeline generates prediction results for the prediction manager from a variety of models optimized for the prediction model optimizer of the analytics pipeline. It is composed of the knowledge DB generator, which generates the knowledge DB.

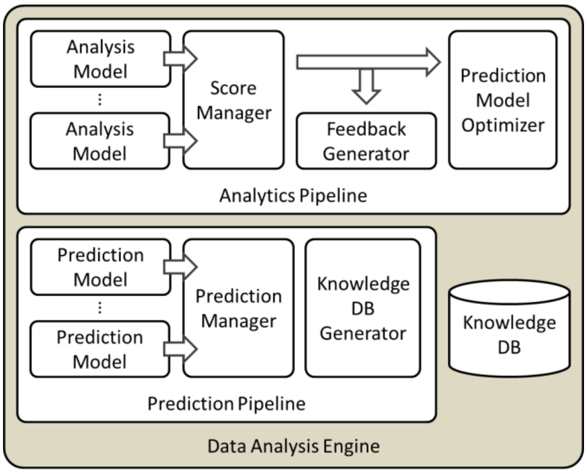


Figure 4. Data analysis engine.

2.4. Service Engine

The service engine, composed of the service layer, performs queries based on the analysis results according to user requirements following a variety of methods. It is composed of a hybrid web/hybrid mobile app that maximizes service through a suitable visualization of the queried analysis results. As shown in Figure 5, the service engine is composed of a user interface manager, which delivers the user requirements to the visualization manager, the visualization manager, which delivers the service manager after a query of user requirements in the knowledge DB, and the hybrid web/app service manager, which enables user service on a web or mobile app.

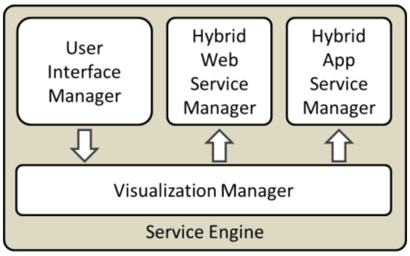


Figure 5. Service engine.

3. Proposed Prediction System Based on Lambda Architecture

The prediction system introduced in this study emphasizes the importance of the real-time data analysis of big data, and it is based on lambda architecture. Lambda architecture has a speed layer that can analyze data generated in real time, addressing the practical problem of general big data analysis pipelines requiring too long an analysis time to match the speed of data generated in real time. It merges the results analyzed in a batch layer, and then provides the analysis results. Figure 6 shows an overview of the proposed prediction system based on lambda architecture for a cardiovascular disorder prediction system.

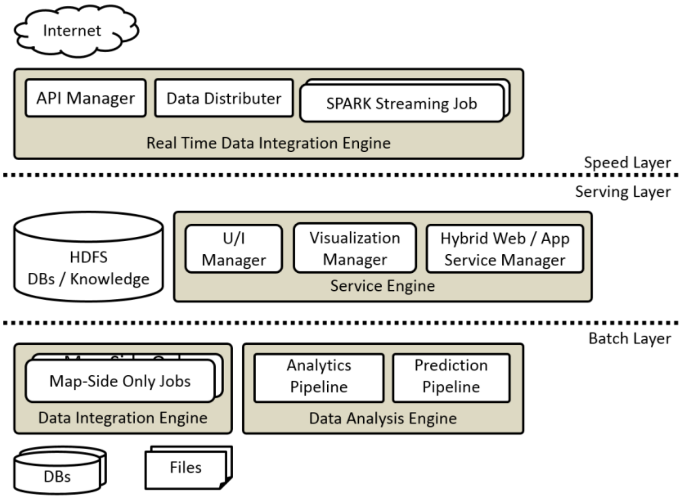


Figure 6. Overview of the proposed prediction system based on lambda architecture. HDFS (Hadoop Distributed File System).

The proposed prediction system with the speed layer in lambda architecture manages the real-time data integration engine in a layer of the same name. This engine delivers the data collected in real time to the data integration engine of the data layer. At the same time, it analyzes the results of the data analysis engine of the analysis layer through the analysis pipeline in real time. The analysis layer performs the analysis for the entire DB, as performed in the batch layer of lambda architecture via the analysis model pipeline, and merges the analysis results delivered from the speed layer. Then, it delivers the results to the prediction model pipeline.

As the data analysis engine merges the analysis results delivered from the speed layer, there should be no redundant results with the analysis results extracted from the analysis model pipeline. To prevent redundancies, the real-time data integration engine of the speed layer applies a time stamp, which can be traced to ensure that these real-time data are indeed recent data that should not be analyzed in the analysis layer.

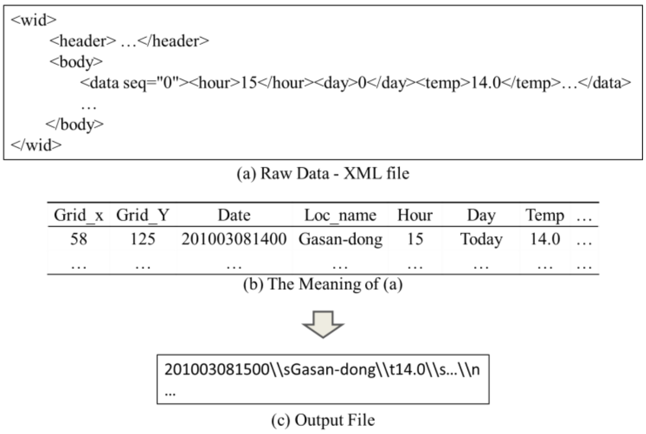
4. Experiment Evaluation

Table 1 shows the example data used in the proposed system. These sample data consist of various file and data types. Preprocessing various types of files can be expensive. The system proposed in this study employs the real-time data integration engine of the speed layer, which processes real-time data, and a data integration engine of the batch layer to process already collected file data. The format of the data processed by each engine is different, but the task is the same. To measure the performance of the proposed system, we evaluated the performance of MapReduce in the batch layer with the highest data preprocessing cost.

Table 1. Sample data of the proposed system.

Data	Type	Size	Provider
Climate Data	API File (XML)	1999~2017	Korea Meteorological Administration
National Health and Nutrition Survey	File (SAV)	1998~2015 60 files (7 GB)	Korea Centers for Disease Control & Prevention
National Core Open Data	API (JSON, XML) File (CSV)	2000~2017 10 files (7 MB)	National Information Society Agency
Health Insurance Statistical Data	API File (XLS)	2001~2017 10 files (7 MB)	Healthcare Bigdata Hub

Figure 7 is an example of climate data preprocessing, where Figure 7 (a) is an xml file of raw data; (b) is a table for improved; and (c) is an example output from the data integration engine. In (b), Grid *x* and Grid *y* represent the parsing of information contained in the <header> tag. Through the preprocessing process, the coordinate value is converted into the position name, and the date is changed to the value reflecting the presentation time at the observation time. Date and location information are converted into key values through the integration process and output, as the result values along with the remaining record data. The output is displayed as the MapReduce key and value outputs; in other words, the key and value are separated by a tab character, the key and value are space characters, and the remaining data are combined.



```
<wid>
      <header> …</header>
      <body>
           <data seq="0"><hour>15</hour><day>0</day><temp>14.0</temp>…</data>
           …
      </body>
</wid>
```

(a) Raw Data - XML file

Grid_x	Grid_Y	Date	Loc_name	Hour	Day	Temp	…
58	125	201003081400	Gasan-dong	15	Today	14.0	…
…	…	…	…	…	…	…	…

(b) The Meaning of (a)

```
201003081500\\sGasan-dong\\t14.0\\s...\\n
...
```

(c) Output File

Figure 7. An example of climate data preprocessing in the batch layer. (**a**) A sample of XML; (**b**) The meaning of the XML file; (**c**) An output file.

Because the MapReduce jobs of the data integration engine are configured as map-side only jobs, it is possible to perform parallel processing on a block-by-block basis. This eliminates Reduce Jobs, thereby preventing performance degradation during data collection. Figure 8 shows a simple experiment of how to design a MapReduce job as a map-side only job, instead of designing it as a MapReduce job. The difference is that MapReduce jobs have no code, and only the mapper's intermediate result with a null key value is output by the dummy reduce, which performs no function. The experiments were performed in stand-alone mode to identify differences in performance, depending on the type of MapReduce job. As shown in Figure 8, as the number of records increases, the performance of the map-side only job increases by 200% compared with the MapReduce job. Because this experiment is performed on a single node, it can be confirmed that the difference in performance increases substantially when the number of nodes increases.

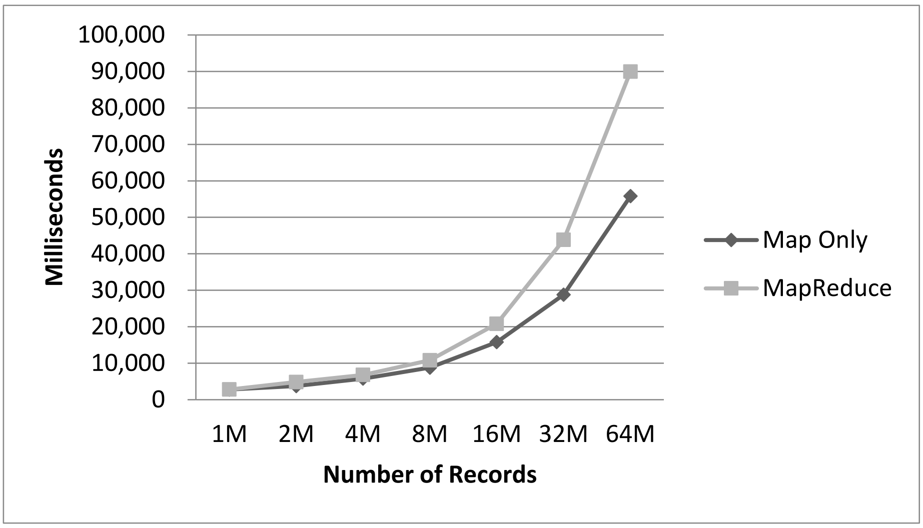


Figure 8. Comparison of the execution time for MapReduce jobs and map-side only jobs according to the number of data records.

5. Conclusions and Future Work

We collected and stored data from a variety of domains that could be factors for cardiovascular disorders and designed a framework to construct a big data integrated DB. The framework was composed of data, speed, analysis, and service layers. The data generated in each layer were stored on distributed storage devices, which includes the national health and nutrition survey provided by the Korea Centers for Disease Control and Prevention, and identical DBs or files containing data provided by the Health Insurance Review and Assessment Service. The data integration engine was implemented as a map-side only job by using the MapReduce framework in Hadoop, thereby adopting the computing power of the nodes comprising the Hadoop cluster in parallel. The real-time data integration engine composed of the speed layer collects real-time data by using APIs, such as the SGIS API of the National Statistical Office or public data APIs. The streaming data analysis pipeline is composed of the data collected by using Apache Spark. The data analysis engine composed of the analysis layer was designed as a pipeline for the prediction model based on the optimized analysis pipeline, which can identify correlations between factors and cardiovascular disorders and analyze the results. The service engine composed of the service layer, which can make a query for a variety of user requirements based on the analyzed results, is composed of a hybrid web/hybrid mobile app to maximize service through suitable visualizations with the query results.

The proposed prediction system, which emphasizes the importance of the real-time data analysis of big data, is based on lambda architecture. Regarding the issue problem whereby big data analysis pipelines cannot typically match real-time speeds, the lambda architecture speed layer enables data analysis in real time; it then merges with the results analyzed in the batch layer to provide improved results. Accordingly, we designed a framework for a cardiovascular disorder prediction system based on lambda architecture. In the future, this system can be used to help predict and optimize diagnosis treatments for serious illnesses such as cardiovascular disorders and can also be applied to a variety of other diseases. Based on this study, a variety of health data from many people can be analyzed, including clinical, genomic, and lifestyle data; however, techniques must be developed that can provide the most suitable personal health solutions. By using integrated medical big data platforms, it may be possible to address the challenge of combatting diseases and identify the prediction, progression, and prognosis of diseases through disease correlations, drug side effects, and genome research.

Acknowledgments: This work was supported by the Industrial Strategic Technology Development Program (10047909; Development of diagnosis and treatment system for respiratory diseases based on bio-chips) funded by the Ministry of Trade, Industry and Energy (MOTIE) of Korea, and the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Science, ICT and Future Planning (NRF-2017R1D1A1B03030157) and by the Korean government (MSIP) (NRF-2015R1A2A2A04004251).

Author Contributions: Sang Hun Han designed the big data analysis system for distributed data processing. Ho Sun Shon and Kyung Ah Kim collected the data and wrote the paper. Eun Jong Cha and Kyoung Ok Kim provided critical insights and discussion. All authors read and approved the final manuscript.

Conflicts of Interest: We declare no conflict of interest with other people or organizations.

References

1. Myers, V.; Broday, D.M.; Steinberg, D.M.; Drory, Y.; Gerber, Y. Exposure to particulate air pollution and long-term incidence of frailty after myocardial infarction. *Ann. Epidemiol.* **2013**, *23*, 395–400. [CrossRef] [PubMed]
2. Lee, B.J.; Kim, B.; Lee, K. Air pollution exposure and cardiovascular disease. *Toxicol. Res.* **2014**, *30*, 71–75. [CrossRef] [PubMed]
3. Newby, D.E.; Mannucci, P.M.; Tell, G.S.; Baccarelli, A.A.; Brook, R.D.; Donaldson, K.; Forastiere, F.; Franchini, M.; Franco, O.H.; Graham, I.; et al. Expert position paper on air pollution and cardiovascular disease. *Eur. Heart J.* **2014**, *36*, 83–93. [CrossRef] [PubMed]
4. Goggins, W.B.; Chan, E.Y.; Yang, C.Y. Weather, pollution, and acute myocardial infarction in Hong Kong and Taiwan. *Int. J. Cardiol.* **2013**, *168*, 243–249. [CrossRef] [PubMed]
5. Lin, H.; An, Q.; Luo, C.; Pun, V.C.; Chan, C.S.; Tian, L. Gaseous air pollution and acute myocardial infarction mortality in Hong Kong: A time-stratified case-crossover study. *Atmos. Environ.* **2013**, *76*, 66–73. [CrossRef]
6. Medina-Lezama, J.; Morey-Vargas, O.L.; Zea-Díaz, H.; Bolaños-Salazar, J.F.; Corrales-Medina, F.; Cuba-Bustinza, C.; Chirinos-Medina, D.A.; Chirinos, J.A. Prevalence of lifestyle-related cardiovascular risk factors in Peru: The PREVENCION study. *Rev. Panam. Salud Publ. Am. J. Public Health* **2008**, *24*, 169–179. [CrossRef]
7. National Information Board. *Personalized Health and Care 2020: Using Data and Technology to Transform Outcomes for Patients and Citizens*; HM Government: London, UK, 2014.
8. Roski, J.; Bo-Linn, G.W.; Andrews, T.A. Creating value in health care through big data: Opportunities and policy implications. *Health Aff.* **2014**, *33*, 1115–1122. [CrossRef] [PubMed]
9. Salari, N.; Shohaimi, S.; Najafi, F.; Nallappan, M.; Karishnarajah, I. An improved artificial neural network based model for prediction of late onset heart failure. *Life Sci. J.* **2012**, *9*, 3684–3689.
10. Vijayashree, J.; SrimanNarayanaIyengar, N.C. Heart disease prediction system using data mining and hybrid intelligent techniques: A review. *Int. J. Bio-Sci. Biotechnol.* **2016**, *8*, 139–148. [CrossRef]
11. Duan, L.; Tang, C.J.; Dong, G.; Yang, N.; Gou, C. Survey of emerging pattern based contrast mining and applications. *J. Comput. Appl.* **2012**, *32*, 304–308. [CrossRef]
12. Abbes, H.; Gargouri, F. Big data integration: A MongoDB database and modular ontologies based approach. *Procedia Comput. Sci.* **2016**, *96*, 446–455. [CrossRef]
13. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Commun. ACM* **2008**, *51*, 107–113. [CrossRef]
14. Eckelman, M.J.; Sherman, J. Environmental impacts of the U.S. health care system and effects on public health. *PLoS ONE* **2016**, *11*, 1–14. [CrossRef] [PubMed]

15. Ministry of Environment. *Health Impact Assessment According to Climate Change Linked with Big Data of National Health Insurance*; Korea Environment Institute: Chungnam, Korea, 2015.
16. Tekiner, F.; Keane, J.A. Big data framework. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 1494–1499.
17. Lambda Architecture. Available online: http://lambda-architecture.net (accessed on 25 November 2017).
18. Kiran, M.; Murphy, P.; Monga, I.; Dugan, J.; Baveja, S.S. Lambda architecture for cost-effective batch and speed big data processing. In Proceedings of the 2015 IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2785–2792.
19. Amazon Web Services. *Lambda Architecture for Batch and RealTime Processing on AWS with Spark Streaming and Spark SQL*; Amazon Web Services Inc.: Seattle, WA, USA, 2015.
20. Zikopoulos, P.; Eaton, C. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*; McGraw-Hill Osborne Media: Berkeley, CA, USA, 2011.
21. Familiar, B.; Barnes, J. Real-time processing using azure stream analytics. In *Business in Real-Time Using Azure IoT and Cortana Intelligence Suite*; Familiar, B., Barnes, J., Eds.; Apress: New York, NY, USA, 2017; pp. 169–226.
22. Marcu, O.C.; Tudoran, R.; Nicolae, B.; Costan, A.; Antoniu, G.; Pérez-Hernández, M.S. Exploring shared state in key-value store for window-based multi-pattern streaming analytics. In Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Madrid, Spain, 14–17 May 2017; pp. 1044–1052.
23. Sideris, K.; Nejabati, R.; Simeonidou, D. Seer: Empowering software defined networking with data analytics. In Proceedings of the International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security, Granada, Spain, 14–16 December 2016; pp. 181–188.
24. Karim, M.R.; Kaysar, M.M. *Large Scale Machine Learning with Spark*; Packt Publishing: Birmingham, UK, 2016.
25. Li, B.; Song, M.; Ou, Z.; Haihong, E. Performance comparison and analysis of yarn's schedulers with stress cases. In Proceedings of the 7th International Conference on Cloud Computing and Big Data, Macau, China, 16–18 November 2016; pp. 93–98.
26. Dolev, S.; Florissi, P.; Gudes, E.; Sharma, S.; Singer, I. A survey on geographically distributed big-data processing using MapReduce. *IEEE Trans. Big Data* **2017**. [CrossRef]
27. Rodrigues, R.A.; Lima, L.A.; Goncalves, S.G.; Mialaret, F.S.; Da Chnha, A.M.; Vieira, L.A. Integrating NoSQL, Relational Database, and the Hadoop Ecosystem in an Interdisciplinary Project involving Big Data and Credit Card Transactions. In *Information Technology—New Generations, 14th International Conference on Information Technology, Las Vegas, NV, USA, 10–12 April*; Springer: New York, NY, USA, 2017; pp. 443–451.
28. Bagwari, N.; Kumar, O. Indexing optimizations on Hadoop. In Proceedings of the 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–7.
29. Kumar, R.; Parasher, B.B.; Gupta, S.; Sharma, Y.; Gupta, N. Apache Hadoop, NoSQL and NewSQL solutions of big data. *Int. J. Adv. Found. Res. Sci. Eng.* **2014**, *1*, 28–36.
30. Gribaudo, M.; Iacono, M.; Kiran, M. A performance modeling framework for lambda architecture based applications. *Future Gener. Comput. Syst.* **2017**. [CrossRef]
31. Decaneto, A. Design and testing of an active big data architecture for social and crowding emergency management. *Politecnico Milano* **2017**. Available online: http://hdl.handle.net/10589/134427 (accessed on 25 November 2017).