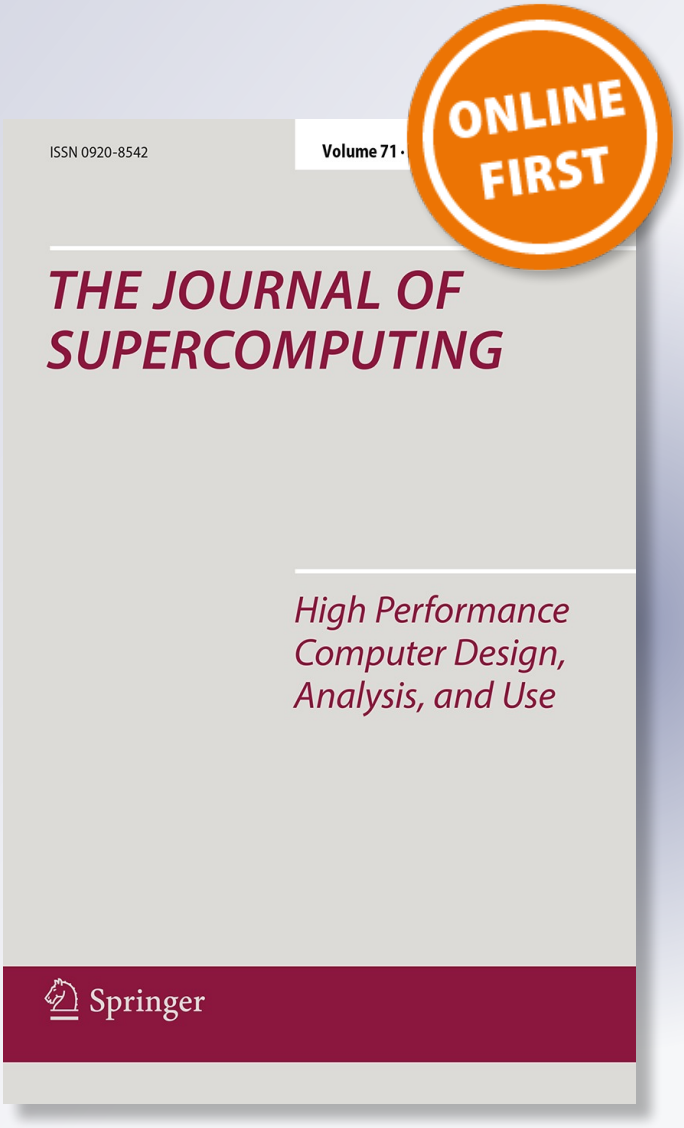
Health Fog: a novel framework for health and wellness applications

Mahmood Ahmad, Muhammad Bilal Amin, Shujaat Hussain, Byeong Ho Kang, Taechoong Cheong & Sungyoung Lee

The Journal of Supercomputing
An International Journal of High-Performance Computer Design, Analysis, and Use

ISSN 0920-8542

J Supercomput
DOI 10.1007/s11227-016-1634-x



Springer

J Supercomput
DOI 10.1007/s11227-016-1634-x

Health Fog: a novel framework for health and wellness applications

Mahmood Ahmad[1] · **Muhammad Bilal Amin**[1] · **Shujaat Hussain**[1] · **Byeong Ho Kang**[2] · **Taechoong Cheong**[1] · **Sungyoung Lee**[1]

Abstract In the past few years the role of e-health applications has taken a remarkable lead in terms of services and features inviting millions of people with higher motivation and confidence to achieve a healthier lifestyle. Induction of smart gadgetries, people lifestyle equipped with wearables, and development of IoT has revitalized the feature scale of these applications. The landscape of health applications encountering big data need to be replotted on cloud instead of solely relying on limited storage and computational resources of handheld devices. With this transformation, the outcome from certain health applications is significant where precise, user-centric, and personalized recommendations mimic like a personal care-giver round the clock. To maximize the services spectrum from these applications over cloud, certain challenges like data privacy and communication cost need serious attention. Following

✉ Mahmood Ahmad
rayemahmood@oslab.khu.ac.kr

✉ Taechoong Cheong
tcchung@khu.ac.kr

Muhammad Bilal Amin
mbilalamin@oslab.khu.ac.kr

Shujaat Hussain
shujaat.hussain@oslab.khu.ac.kr

Byeong Ho Kang
Byeong.Kang@utas.edu.au

Sungyoung Lee
sylee@oslab.khu.ac.kr

1 Ubiquitous Computing Lab, Department of Computer Engineering, Kyung Hee University, Seoul, Korea

2 School of Computing and Information Systems, University of Tasmania, Hobart, Australia

Published online: 29 January 2016

the existing trend together with an ambition to promote and assist users with healthy lifestyle we propose a framework of Health Fog where Fog computing is used as an intermediary layer between the cloud and end users. The design feature of Health Fog successfully reduces the extra communication cost that is usually found high in similar systems. For enhanced and flexible control of data privacy and security, we also introduce the cloud access security broker (CASB) as an integral component of Health Fog where certain polices can be implemented accordingly. The modular framework design of Health Fog is capable of engaging data from multiple resources together with adequate level of security and privacy using existing cryptographic primitives.

Keywords E-health and wellness applications · Big data · IoT · Cloud storage · Fog computing · Cloud access security broker

1 Introduction

In the past few years, health and wellness applications have emerged as a fast growing category of mobile applications. This increasing trend is considered as a prompt and useful resource for collecting users' data which are used for generating recommendations for a healthy lifestyle. Using smart phone features, applications like Microsoft Health, Apple Healthkit, Samsung S Health, and Google Fit collect users data by monitoring their daily activities, e.g., eating habits, sleeping patterns, and workout routines to generate certain recommendations which are helpful in maintaining a healthy lifestyle. To expand the spectrum of these recommendations, the data acquired from smart phones can be further synergized with other data resources like wearable sensors and a smart home environment. The processing on this integrated data acquired from various resources encompasses comprehension towards overall recommendations, thus creating the favorable environment to further engage other possible data resources including, but not limited to, social media and personal health records too. This expansion on data intake from various resources enables health and wellness applications to advise personalized and user-specific recommendations rather than giving general tips for a healthy lifestyle. Due to this reason the adaptation rate of such applications is on the rise with downloads in millions [1–4]. These applications offer a variety of features and plans like weight-loss, calorie-counter, women-health, and activity-recognition. To maximize the feature space of an application there is another trend of data-cross-sharing in which one application can share its data with the other, e.g., Fitocracy [5] can share its data with the RunKeeper [1].

The amount of data generated by smart phones and supportive need to include data from other resources make data volume enormous and its structure more complex. Although smart phones are sufficiently equipped with large memory size and computational resources for on-device storage and processing ability, however, to achieve increased battery life, data backup, centralized data storage, and to fulfill data-cross-sharing, there is another approach gaining momentum in a majority of applications which is the adoption of cloud services. Applications that used to store

Features / Systems	Data Source						Storage & Security				UI/UX			Services						Information Sharing			Knowledge Maintenance		
	Sensory Data	User profile	IOT	Other apps	Clinical data	Social media	User device	Cloud storage	Big data storage	Encrypted storage	User Experience	User Modeling	Adaptation of UI	Activity Recognition	Expert Svc	Wellness svc	Personal recommendation	Clinical svc	SDK/API	With other apps	Social media	Other users	Open Knowledge	Knowledge acquisition	Knowledge evolution
Google Fit	☑	☑	☑	☑	☒	☒	☑	☑	☑	☑	☒	☑	☒	☑	☑	☑	☒	☒	☑	☑	☑	☑	☑	☑	☒
Samsung S health	☑	☑	☑	☑	☒	☒	☑	☑	☑	☑	☒	☑	☒	☑	☒	☑	☑	☑	☑	☑	☑	☑	☑	☑	☒
Microsoft health	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☒	☑	☒	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Apple healthkit	☑	☑	☑	☑	☒	☒	☑	☑	☒	☑	☒	☑	☒	☑	☑	☑	☑	☒	☑	☑	☑	☑	☒	☑	☒
Open mhealth	☑	☑	☑	☑	☒	☒	☑	☑	☒	☒	☒	☑	☒	☑	☑	☑	☑	☒	☑	☒	☑	☑	☒	☑	☒
NoomCoach	☑	☑	☑	☑	☒	☒	☑	☑	☒	☒	☒	☑	☒	☑	☑	☑	☑	☒	☒	☑	☑	☒	☒	☑	☒
Argus	☑	☑	☑	☒	☒	☒	☑	☑	☒	☑	☒	☑	☒	☑	☑	☑	☒	☒	☒	☒	☑	☒	☒	☑	☒
Runtastic	☑	☑	☑	☑	☒	☒	☑	☑	☑	☑	☒	☑	☒	☑	☑	☑	☑	☒	☑	☑	☑	☒	☑	☑	☒
Runkeeper	☑	☑	☑	☑	☒	☒	☑	☑	☑	☑	☒	☑	☒	☑	☑	☑	☒	☒	☑	☑	☑	☒	☑	☑	☒
Zombie Run	☑	☑	☒	☒	☒	☑	☑	☑	☒	☒	☒	☑	☒	☑	☒	☑	☒	☒	☒	☒	☑	☒	☒	☑	☒
Mining Mind	☑	☑	☑	☒	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☒	☒	☑	☑	☑	☑	☑

Fig. 1 Health and wellness applications: features survey and their categorization

data locally are swiftly moving towards centralized data storage through the cloud. Using cloud services, applications can conveniently afford inclusion of data from multiple resources and subsequent execution of substantial and complex computations. In addition, the application access also become more flexible and ubiquitous in comparison to the legacy approach where only a single device has the privilege to exploit application features on which it is installed. A user can now access these applications more seamlessly on a range of devices from mobile devices to fixed terminals and even across various applications. The problems of data loss in case of malfunctioning or stolen device is an allied advantage of using cloud services for convenient recovery.

Mobile health applications are being designed to promote fitness and track beneficial health metrics aiding healthy living. To monitor and track daily activities there is a huge rack of applications with variety of services. After selecting few of them, we organize and very briefly explain their features in certain groups to analyze growth and trend of offered services as shown in Fig. 1.

1.1 Data source

The integrated sensors within a smart phone are the initial and foremost sources for data acquisition. For personalized recommendations the sensory information is fused together with user profile like her age, gender, BMI index, weight, food preferences, and disability information. In addition, many applications also acquire and utilize data generated by physical devices including wearables and IoT (internet of things) like pedometer, heart rate monitor, surveillance camera, and smoke detector. The social media and clinical data are a gradual and influential addition being considered while generating the user-centric recommendations.

1.2 Storage and security

The data generated by health applications are usually stored on the user device. While opting for on-device storage, majority of applications lacks backup strategy of data and also do not employ encrypted data storage, thus considering physical possession of a device sufficient against data protection. This assumption of protection fails when a user is deprived of her device and loses it due to some reason. In this situation the device offers same privileged data access to any unauthorized user having the device possession. Also, the same situation leaves no alternate option for data recovery as well. Due to these reasons and realizing the necessity of protecting the applications data, its storage is deemed essential with encrypted storage along with opting for a storage location where data can be restored conveniently. Realization of these facts introduce the pivotal role of cloud computing as a growing trend amongst health applications. With ever increasing sophistication and integrated sensors in smart phones, health applications create enormous amount of data that are now more convenient to store and process with the induction of cloud. With this convenience, health applications can easily afford to engage and process data which are generated by other devices (IoT), wearables and third-party applications along with the data that are generated indigenously. This homogenization expands and evolves data into big data as current era of health applications deals with more volume, variety, and velocity of data used ever before. With this transformation, the data-driven health applications can now generate more realistic and personalized recommendations in comparison to legacy approach of generating general recommendation due to limited input of available data.

1.3 User interface (UI)/ user experience (UX)

The user interface (UI) has an important and profound role between the application and its user. The information on user interaction can be collected through user interface to maximize user experience. The importance of this aspect in modern applications is due to the fact that static interface lacks the ability to reflect user needs and satisfaction. In certain situations the interface has to be redesigned and adjusted with respect to specific needs of its users, e.g., increased font size for a user with week eyesight. In this regard, the information from user profile data (age, gender, week eyesight or any other disability) together with the contextual information (mode, time of the day, weather, etc.) can be used for customizing the application interface. The application designed on these features adapts itself according to the situation and makes user experience more comfortable while interacting with the application. In addition, involving user feedback through continuous monitoring of how she interacts with the application adds more refinement and perfection for enhanced user experience.

User experience is an evolutionary process to adapt and personalize the user interface. The personalization aspect is the most important factor and is achievable with user involvement where interface is dynamic and subject to user experience. The adaptive UI is managed by the UX, user feedbacks, and movement in the application which is then fused with the user profile data. The user movements encompass number of clicks, color schemes, text size, brightness and navigation within the application. All these

parameters can be used in a user satisfaction module where suitable weight assignment is done for the collected parameters. The accumulated weight along with certain bounds in terms of threshold can be shared and further verified with domain expert. A similar system has been proposed by Hussain et al. [30] in which user feedback, web monitoring, and contextual information are used for adaptive UI with the help of UX.

1.4 Services

Majority of applications are capable of recognizing user activities (walking, jogging, running, sleeping) and present this information visually at the end of the day or whenever required by the user. Other than presenting the activities individually and their time duration, few applications also present the impact of one activity over the other, e.g., sleep efficiency with number of steps taken or changing heart rate due to running [3]. To promote the healthy lifestyle this information is also used to predict and anticipate certain trends, e.g., weight, calories deposit, or sleep efficiency. Sharing this information with health experts adds more comprehension and perfection towards user wellness with personalized recommendations. Clinical services and provisioning of SDK/API are also potential and growing features offered by few applications.

1.5 Information sharing

The data-driven landscape for health and wellness applications has revolutionized their core operations by expanding the boundaries for data/information sharing across various applications. Instead of making data silos as an independent island of information, its sharing is now more frequent involving other applications, devices, social media, and domain experts. The application of Fitcoracy [5] can sync with RunKeeper [1], DigifitIcardio [6] can use data generated by a heart-rate monitoring device Mio [7]. Similarly, users can share their data through social media within a community of their peers [4] and also with domain experts, e.g., clinicians, personal care-givers, and dietitians.

1.6 Knowledge maintenance

The knowledge which is derived from the personalized recommendations, user satisfaction level, and feedback analysis is preserved for later use as well as shared as an open knowledge. The discussion about knowledge acquisition, maintenance, and evolution in health applications is beyond the scope of this paper; therefore, it is just highlighted as a feature alone.

Figure 1 represents growing trend within health and wellness applications with respect to data and their utility. The importance of diverse data sources and induction of public cloud is obvious; however, it also introduces significant challenges of data heterogeneity resolution, implications associated with personal data on public cloud, and secure sharing of user data and information with other entities. With the ambition of promoting healthy living and considering the aforementioned challenges,

we propose a framework aiming swift and secure services. The proposed framework employs the Fog Computing as a mediator layer between the system entities used for data acquisition and system consumers over public cloud. The rational behind using the Fog Computing is to bring computational resources close to data generating entities and preserving privacy aspect of users data during heterogeneity resolution of multivariate data. Considering data preprocessing as a demanding task with respect to computation and energy, it is not feasible to perform it on the user device; also, executing the same on public cloud can expose individual identity thus compromising individual identity with information exposure. These end-to-end limitations can be resolved with data preprocessing within Fog. At the same time, only relevant and useful information is uploaded into the public cloud after preprocessing, thus avoiding unnecessary communication overhead due to entire upload of data. In this paper we make following contributions:

- computational task for data heterogeneity resolution is done through Fog Computing[1] instead of public cloud, thus minimizing the information leakage during this process,
- services pool and data access policies and guidelines are staked in Health Fog giving more flexibility in terms of management and autonomous control,
- The processed and encrypted data are made available to authorized users obliviously using existing cryptographic standards.

2 Definitions and technical preliminaries

Before describing the HealthFog framework, we present some definitions and technical preliminaries.

2.1 Fog-computing

Fog computing, a Micro Datacenter paradigm, is a highly virtualized platform, which provides computation, storage, and networking services between the end nodes in an internet of things (IoT) and traditional clouds [8]. In contrast to the cloud, which is more centralized, Fog computing targets the services and applications with widely distributed deployments. Fog is aimed to deliver high-quality streaming to mobile nodes, like moving vehicles, through proxies and access points positioned accordingly, like, along highways and tracks. Fog suits applications with low latency requirements, emergency and healthcare-related services, video streaming, gaming, augmented reality, etc.

2.2 Cloud access security broker

Cloud Access Security Brokers (CASB) are quickly emerging as a must-have security solution for organizations looking to adopt cloud-based applications. CASBs are either on-premise, or cloud-based (or both) security policy enforcement points which are

[1] Depending upon the nature of proposed framework we will refer Fog computing as Health Fog.

placed between end users and the various cloud service providers. CASBs can inspect traffic, alert on anomalous behavior, and in most cases provide some level of data loss prevention (DLP) enforcement. Cloud Access Security Brokers can also consolidate multiple types of security policy enforcement, e.g., user authentication, single sign-on, authorization, credential mapping, device profiling, encryption, tokenization, logging, alerting, and malware detection/prevention [9,10].

2.3 Homomorphic encryption

Homomorphic encryption *HE* is a form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another (possibly different) algebraic operation performed on the ciphertext. An encryption scheme is said to be additive homomorphic if and only if

$$E_H(m_1) \odot E_H(m_2) = E_H(m_1 + m_2),$$

where $\odot$ is an operator. Pascal Paillier cryptosystem [23] possesses the property of additive *HE* which is as follows:

- Key generation: Let $N = pq$ be the RSA-modulus and g be an integer of order αN module N^2 for some integer α. The public key is (N, g) and the private key is $\lambda(N) = lcm((p-1)(q-1))$.
- Encryption: The encryption of message $m \in Z_N$ is $E_h(m) = g^m r^N \bmod modN^2$ where $r \in_R Z_N^*$
- Decryption: For ciphertext c, the message is computed from

$$m = \frac{L(c^{\lambda(N)} modN^2)}{L(g^{\lambda(N)} modN^2)}$$

A scheme is said to be multiplicative homomorphic if and only if

$$E_H(m_1) \odot E_H(m_2) = E_H(m_1 \times m_2)$$

The Goldwasser-Micali (GM) cryptosystem is a semantically secure scheme based on the quadratic residuosity problem. It has XOR homomorphic properties, in the sense that $E_H(b).E_H(b') = E(b \oplus b') modN$, where b and b' are bits and N is the public key. A homomorphic encryption is said to be semantically secure if $E(H)$ reveals no information about m_1 and m_2; hence it is computationally infeasible to distinguish between the cases $m_1 = m_2$ and $m_1 \neq m_2$ [22].

Here is an example of how a homomorphic encryption scheme might work in cloud computing.

2.3.1 Example

Let us assume that a sensitive information comprising number 5 and 10 is encrypted and uploaded in the public cloud. For simplicity and understanding purpose, the corre-

sponding ciphertext of 5 and 10 appears as 10 and 20 after applying the homomorphic encryption, i.e., the algorithm multiplies original values with 2. To perform any operation on these encrypted values the cloud will use 10 and 20, without knowing the original values. To utilize the computational services of cloud for addition purpose, it will use the features of homomorphic encryption and return the answer 30. On receiving end, the value of 30 will be decrypted as 15 using the decryption key.

2.4 Private comparison

Yaos classical millionaires problem involves two millionaires who wish to know who is richer. However, they do not want to find out inadvertently any additional information about each other's wealth. More formally, given two input values x and y, which are held as private inputs by two parties Alice and Bob, respectively. the problem is to securely evaluate the Greater Than (GT) condition through a predicate function f such that $f(x, y) = 1$ if and only if $x > y$, without exposing inputs. We used Fischlin protocol [11] for the private comparison because it allows comparing two ciphertexts encrypted with the GM cryptosystem using the same public key. Fischlin uses the GM-encryption scheme to construct a two-round GT protocol. The GM encryption scheme has the XOR, NOT, and re-randomization properties. They modified the scheme to get an AND property, which can be performed only once. The computation cost $O(n)$ for the server side is very efficient. Nevertheless, the overall computation cost for both the client and server sides are $O(nlogN)$, where N is the modulus. The scheme is as follows:

- Key generation: Let $N = pq$ be the RSA-modulus and z be a quadratic non-residue of Z_n^* with Jacobi symbol +1. The public key is (N, z) and the secret key is (p, q).
- Encryption: For a bit b, the encryption is $E(b) = z^r r^2 \bmod N$, where $r \in_R Z_N^*$.
- Decryption: For a ciphertext c, its plaintext is 1 if and only if c is a quadratic non-residue. If c is a quadratic residue in Z_N, c is quadratic residue in both Z_p^* and Z_q^*.
- xor-property: $E(b_1)E(b_2) = E(b_1 \oplus b_2)$.
- Not-property: $E(b) \times z = E(b \oplus 1 = E(\bar{b})$.
- Re-randomization: Randomization of ciphertext c can be done by multiplying an encryption of 0.

3 Proposed system overview

The proposed system is designed in a layered architecture comprising data generating entities DGE, Fog computing and end users. Hospitals and clinical institutes, smart home environment, and users equipped with wearables and smart-phone are main entities of DGE. For brevity, the following discussion is with respect to a single user 'Alice' who is a member in DGE and formally expressed as DGE_{Alice}. Alice is living in a smart home environment and using few wearables to monitor her daily activities as shown in Fig. 2. Her passion to achieve a healthy lifestyle is triggered

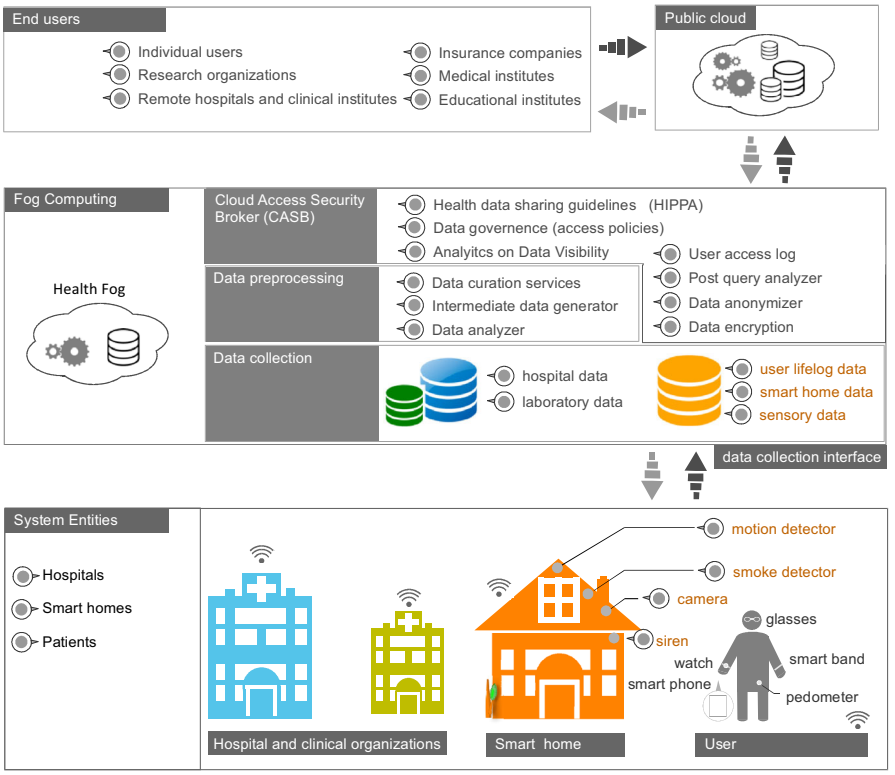


Fig. 2 Health Fog: system architecture

due to a recent discovery on her weight gain during a visit to a nearby hospital. The preliminary diagnosis highlights the sedentary life style caused by irregular routines for diet and sleep along with inadequate physical workout. The doctor advises her to change monotonous routines causing weight abnormality by inducing sufficient workout and that too with regular feedback and careful monitoring round the clock. Considering the high tendency of weight gain in her family tree, she also decides to avail expert services from a nutritionist for appropriate diet selection. To apprise the doctor on her lifestyle and to seek specialized advise from a domain experts, a mechanized system is required that should support data shareability with adequate security and high availability. With these requirements, Alice joins the Health Fog where hospitals, domain experts, and other people like her are already registered, collaboratively pursuing a healthy lifestyle.

4 Data outsourcing

Hospital and a user are two main data generating entities referred as DGE_H and DGE_U. A user living in a smart home environment using various devices d_i is represented as $DGE_U \langle d_1, d_2, \ldots, d_n \rangle$. For any device d_i that is required to be used as

an active member of DGE_U is commissioned with Health Fog through Health Fog client stub. A client stub is an application running on a smart phone responsible for data uploads, activation or deactivation of a particular device, i.e., $d_i \in DGE$. Provisioning of activation and deactivation of a commissioned device is meant to control data uploads from user's side as per individual needs and concerns of privacy. The client stub is also responsible for providing end to end user services through personalized recommendations, alerts, statistical analysis on daily activities, and expert advice. Other than this independent link established between the client stub and the Health Fog, DGE_H also deposits its data by its own and separate link with the Health Fog. The data which are uploaded by the DGE_H are assumed to be preprocessed with the removal of identifiable information of a patient/visitor of that hospital.

5 Health Fog

The data arriving at the data collection layer of Health Fog are preserved under defined categories for user $DGE_U^{\circledast}$ and hospital $DGE_H^{\circledast}$. The data are then preprocessed for any missing value, noise reduction, erroneous values, duplicates, correct data labeling, and reduced as $DGE_U^{\odot}$ and $DGE_H^{\odot}$. The process Δ of data transformation for DGE_U and DGE_H is defined as

$$\begin{aligned} DGE_U^{\circledast}{}_{\circledast} &\xrightarrow{\Delta_{\amalg}^{\theta}} DGE_U^{\odot} \\ DGE_H^{\circledast}{}_{\circledast} &\xrightarrow{\Delta_{\amalg}^{\theta}} DGE_H^{\odot} \end{aligned} \tag{1}$$

Through process Δ, we claim two advantages of data reduction and privacy gain represented as $\Delta_{\amalg}$ and Δ^{θ}, respectively. Collectively these two operations are represented as $\Delta_{\amalg}^{\theta}$ in Eq. 1. With data reduction we mean the transformation of raw data into useful information, whereas the privacy gain is the information exposure of knowing individual identity while processing the raw data. Since these two operations are performed within Health Fog, communication overhead and information exposure for a curious cloud are marginally reduced which could appear otherwise.

The role of Cloud Access Security Brokers (CASB) is aimed to improve Health Fog security features by placing it in-between the public cloud and its consumers. CASB helps Health Fog to improve its monitoring, visibility, and control of user and data activity on public cloud. It also ensures that unauthorized parties do not gain access to corporate resources in the cloud by unification of consolidated polices ensuring their consistency and effectiveness at all operational frontiers. With CASB we propose following services.

5.1 Cloud access security broker (CASB)

The personal data which have to be shared through public cloud need to be in compliance with certain rules and regulations. The statutory body of these rules and regulations is accommodated within the CASB. For this purpose we refer to HIPPA as one of the policy guidelines while making data availability for the sharing purpose. Besides, polices that are defined at an organizational level can also be made part of

CASB to further meet local needs for data sharing and its access. Through CASB the visibility into user data and access is also provisioned through analytic and data visibility sub component. For each access that passes through CASB, it is logged for post analysis for anomaly detection in user query or data access pattern. The query log for post query analysis is used for discovering the data access trends. The access trends helps to improve data provisioning both in terms of refinement and improved security. To protect the individual identity from outsourced data without encryption, the anonymization is an option that can be used instead. With anonymization that involves techniques like generalization, suppression, and perturbation, to hide individual's identity is an option for data sharing without encryption, thus making data utility aspect more flexible with large number of users requiring no authentication. Although with anonymization the utility aspect on data becomes relatively minimal, however, it saves the cost of encryption/decryption and user management. For more details on anonymization readers may refer to [12,13].

The security and privacy issues on outsourced data in public cloud are not limited with the nature of data alone but also involve encryption techniques and access model as well. While achieving desirable security and privacy features on outsourced data, encrypted outsourcing in public cloud is highly recommended [14–16]. For this reason the data outsourcing from Health Fog into the public cloud is provisioned with encryption. Addressing the general issues while interacting with the encrypted data over cloud mainly includes user management and how the data are accessed or explored. With exploration, we refer retrieving the desired results from encrypted data against user query parameters. In this regard, the user access is controlled with the help of authorized credentials and a decryption key. In Health Fog, the process of user verification is done through CASB followed by request routing to public cloud where data are outsourced. The request arriving at cloud is then processed against the request parameters over encrypted data. Besides protecting the outsourced public data, protecting request parameters is equally important and emphasized. After processing the user query the result is sent back to the user where it is decrypted with authorized key. This whole process for data outsourcing, user accession, and response extraction is given in the following subsections.

5.1.1 Setup

Let $DGE_U^{\circledast}$ and $DGE_H^{\circledast}$ b the raw data collected from the user and hospital, respectively. After going through the process of Δ, these data are collectively represented as $D_*^{\odot}$

$$DGE_U^{\odot} + DGE_H^{\odot} = D_*^{\odot} \tag{2}$$

For quick data exploration on $D_*^{\odot}$, an index $\mathcal{I}$ is also created for all unique keywords against the outsourced data. Once $\mathcal{I}$ is generated, Health Fog initializes proxy re-encryption by generating Health Fog key (ω_o), user key (ω_u), and transformation key ($\omega_{o\rightarrow u}$). The owner key (ω_o) ensures the privacy of keywords within $\mathcal{I}$, whereas keyword frequencies are concealed with CASBs secret key (sk). The Health Fog key (ω_o) ensures the privacy of keywords within $\mathcal{I}$. The user key (ω_u) is used by the user to encrypt search criteria. The Health Fog only shares (ω_o) with the authorized users.

The transformation key $(\omega_{o\to u})$ is used by the cloud server to transform ciphertext (encrypted inverted index). Transformation of ciphertext ensures that the Health Fog does not need to outsource separate encrypted index for each authorized user. Each user is also provided with authorized credentials, i.e., user id and password that are used each time a user forwards her request through Health Fog.

5.1.2 Data outsourcing

For privacy aware data processing and oblivious request evaluation of user query on cloud, Health Fog encodes $\mathcal{I}_{kw_{0\ldots n}}$ using a publicly known encoding function denoted as $\mathcal{H}$, i.e., $\mathcal{H}\left(\mathcal{I}_{kw_{0\ldots n}}\right) \rightarrow \hat{\mathcal{I}}_{kw_{0\ldots n}}$. The encoded keywords $\left(\hat{\mathcal{I}}_{kw_{0\ldots n}}\right)$ are then encrypted with proxy reencryption algorithm using $\mathcal{E}_p\left(\hat{\mathcal{I}}_{kw_{0\ldots n}}\right) \rightarrow \hat{I}^{\omega_o}_{kw_{0,\ldots n}}$. To ensure that the cloud server cannot learn any information from the inverted index, Health Fog encrypts $\mathcal{I}_{f_{0,\ldots,n}}$ with CASB secret key, i.e., $\mathcal{E}\left(\mathcal{I}_{f_{0,\ldots,n}}, sk\right) \rightarrow \mathcal{I}^{sk}_{f_{0,\ldots,n}}$. After that, Health Fog encrypts (ω_u) with the public key of the user to whom it wants to grant searching capabilities over the outsourced data, i.e. $\mathcal{E}\left(\omega_u, k_{pub}\right) \rightarrow \omega_u^{k_{pub}}$. In a cloud storage system, outsourced data can be shared with multiple users each having its own access privileges over the outsourced data. With proxy reencryption Health Fog does not need to encrypt $\mathcal{I}_{kw_{0\ldots n}}$ separately to permit each authorized user to query $\hat{\mathcal{I}}^{\omega_o}_{kw_{0\ldots n}}$. An authorized user can submit its query encrypted with its proxy reencryption secret key $\left(\omega_{u_i}\right)$. Cloud server then transforms $\hat{\mathcal{I}}^{\omega_o}_{kw_{0\ldots n}}$ to $\hat{\mathcal{I}}^{\omega_{u_i}}_{kw_{0\ldots n}}$ using an appropriate transformation key $(\omega_o \rightarrow u_i)$ provided by the Health Fog. Thus, the Health Fog only needs to encrypt $\hat{\mathcal{I}}^{\omega_o}_{kw_{0\ldots n}}$ once, and n authorized users can query it, without compromising privacy of the outsourced data.

5.1.3 Query generation

In order to privately search the cloud storage, a user obtains its proxy reencryption secret key from the Health Fog and deciphers it using the private key, i.e. $\mathcal{D}\left(\omega_u^{pub}, k_{pri}\right) = \omega_u$. The user then defines a search criteria $\left(\mathcal{C}_{kw_{0,\ldots l}}\right)$ that consist of a list of keywords $k_{w_0}, \ldots, k_{w_l}$. Then $\mathcal{C}_{kw_{0,\ldots l}}$ is encoded using a publicly known encoding function, i.e. $\mathcal{H}\left(\mathcal{C}_{kw_{0,\ldots l}}\right) \rightarrow \hat{\mathcal{C}}_{kw_{0,\ldots l}}$, where $\mathcal{H}$ is the same as used by the Health Fog during data outsourcing. To ensure confidentiality of the keywords, $\hat{\mathcal{C}}_{kw_{0,\ldots l}}$ is encrypted with proxy reencryption using the proxy reencryption secret key, i.e. $\mathcal{E}_p\left(\hat{\mathcal{C}}_{kw_{0,\ldots l}}, \omega_u\right) = \hat{\mathcal{C}}^{w_u}_{kw_{0,\ldots l}}$.

Once privacy of the search criteria is assured, it is send to CASB who uses it to model oblivious search query. On receiving $\hat{\mathcal{C}}^{w_u}_{kw_{0,\ldots l}}$ the CASB defines a polynomial $(P(x))$, such that each element of $\hat{\mathcal{C}}^{w_u}_{kw_{0,\ldots l}}$ is a root of $P(x)$, i.e. $P(x \in \hat{\mathcal{C}}^{w_u}_{kw_{0,\ldots l}}) = \sum_{i=0}^{l} \alpha_i x^i = 0$.

Once $P(x)$ is defined in accordance with $\hat{\mathcal{C}}^{w_u}_{kw_{0,\ldots l}}$, the CASB then initializes homomorphic encryption by generating a public key $\left(\sigma_{pk}\right)$ and secret key (σ_{sk}). The CASB encrypts the coefficients $\left(\alpha_{o,\ldots l}\right)$ of $P(x)$ with homomorphic encryption algorithm using σ_{sk}, i.e. $\mathcal{E}_H\left(\alpha_{0,\ldots l}, \sigma_{sk}\right) = \alpha^{\sigma_{sk}}_{0,\ldots l}$. Subsequently $\alpha^{\sigma_{sk}}_{0,\ldots l}$ and σ_{pk} are transferred to

the cloud server. Encrypted coefficients $\alpha^{\sigma_{sk}}_{0,\ldots l}$ are used to execute search query over encrypted inverted index $\left(\hat{\mathcal{I}}^{\omega_o}_{kw_{0,\ldots,n}}\right)$. In the context of search over encrypted data, set intersection can be used to execute search query by matching search criteria with the inverted index.

5.1.4 Searching

Cloud server hosts the encrypted inverted index as encrypted keywords $\left(\hat{\mathcal{I}}^{\omega_o}_{kw_{0,\ldots,n}}\right)$ and their concealed frequencies $\mathcal{I}^{sk}_{f_{0,\ldots,n}}$ along with the outsourced data $\mathcal{F}$. Encrypted query $\alpha_{0,\ldots l}$ submitted by the CASB is evaluated against $\hat{\mathcal{I}}^{\omega_o}_{kw_{0,\ldots,n}}$. On receiving $\alpha^{sk}_{0,\ldots l}$, cloud server transforms $\hat{\mathcal{I}}^{\omega_o}_{kw_{0,\ldots,n}}$ with the respective users transformation key $\omega_{o\rightarrow u}$, provided by the Health Fog, i.e. $\mathcal{T}_p\left(\hat{\mathcal{I}}^{\omega_o}_{kw_{0,\ldots,n}}, \omega_{o\rightarrow u}\right) \rightarrow \hat{\mathcal{I}}^{\omega_u}_{kw_{0,\ldots,n}}$. Once the encrypted index is transformed, cloud server defines a polynomial $P(y)$, using each element of $alpha_{0,\ldots l}$ as a coefficient of $P(y)$. It then computes oblivious value $\left(\Delta_{y_i}\right)$ by evaluating $r.P(y_i)$, where $y_i \in \hat{\mathcal{I}}^{\omega_u}_{kw_{0,\ldots,n}}$ and r is a random number, i.e. $\Delta_{y_i} = r.P(y_i)$.

As the query is concealed using homomorphic encryption, cloud server cannot learn any information from $P\left(y_i \in \hat{\mathcal{I}}^{\omega_u}_{kw_{0,\ldots,n}}\right)$. Once the cloud server has evaluated $P\left(y_{0,\ldots,n} \in \hat{\mathcal{I}}^{\omega_u}_{kw_{0,\ldots,n}}\right) = \Delta_{y_{0,\ldots,n}}$, it replies back the query evaluation resultlist of oblivious values along with the concealed keyword frequencies to the CASB, i.e. $\Delta_{y_{0,\ldots,n}}, \mathcal{I}^{sk}_{f_{0,\ldots,n}}$.

5.1.5 Response extraction

On receiving the cloud server response $\Delta_{y_{0,\ldots,n}}, \mathcal{I}^{sk}_{f_{0,\ldots,n}}$, CASB decrypts the oblivious values using the homomorphic secret key, i.e. $\mathcal{D}_H\left(\Delta_{y_i}, \sigma_{sk}\right) = \psi_i$, where ψ_i can be zero or a random number. As the search query was modeled as a polynomial having roots equal to the concealed search criteria, i.e. $P(x \in \hat{\mathcal{C}}^{\omega_u}_{kw_{0\ldots l}}) = \sum_{i=0}^{l} \alpha_i x^i$, query evaluation at cloud server can result either in a zero or a non-zero value shown in Eq. 3.

$$P(y_i) = \begin{cases} \psi_i = 0 \; if \left\{ y_i | y_i \in \hat{\mathcal{I}}^{\omega_u}_{kw_{0,\ldots,n}} \wedge y_i \in \hat{\mathcal{C}^{\omega_u}_{kw_{0,\ldots,l}}} \right\} \\ \psi_i = 0 \; if \left\{ y_i | y_i \in \hat{\mathcal{I}}^{\omega_u}_{kw_{0,\ldots,n}} \wedge y_i \notin \hat{\mathcal{C}^{\omega_u}_{kw_{0,\ldots,l}}} \right\} \end{cases} \tag{3}$$

Zero value reveals that inverted index contains keyword that matches with the concealed search criteria specified by the user, i.e. $\hat{\mathcal{C}}^{\omega_u}_{kw_i} \in \hat{\mathcal{I}^{\omega_u}_{kw_{0\ldots n}}}$, whereas non-zero reveals that concealed search criteria do not match with any of the keyword in inverted index; consequently, CASB recovers r. Once encrypted keywords are identified, CASB deciphers the corresponding frequency index using the secret key, i.e.$\mathcal{D}_S\left(\mathcal{I}^{sk}_{fi}, sk\right) \rightarrow \mathcal{I}_{fi}$. After sorting the encrypted keywords based on their frequency count, CASB replies oblivious results to the user.

On receiving the CASBs response, a user deciphers the search criteria using its proxy reencryption secret key. Through decryption the user learns the keyword that matches

with the encrypted index, i.e. $\mathcal{D}_P\left(\hat{\mathcal{C}}^{\omega_u}_{kw_{0\ldots k}}, \omega_u\right) = \hat{\mathcal{C}_{kw_{0\ldots k}}}$, where k is the number terms that are identical between $\hat{\mathcal{C}_{kw_{0\ldots l}}}$ and $\hat{\mathcal{I}^{\omega_u}_{kw_{0\ldots n}}}$. During the query evaluation cloud server learns nothing about the inverted index or the search criteria, however; it accurately evaluates the search query and replies back the oblivious response.

6 Evaluation and results

For experiment purpose we have used the Samsung Galaxy S-III smart phone. The sensory input of accelerometer and GPS has been used for activity detection and subsequent recommendations by the domain experts, i.e., doctor and the nutritionist. The data generated by these sensors of smart phone are retrieved on a 3-s interval. For optimal communication between the client stub and Health Fog, the 3-s interval data are batched over a minute and sent to the Health Fog after 60 s. For activity detection, these data are deposited on Health Fog through client stub along with other sensory inputs collected from the smart home environment. The user data generated within the hospital like history, prescriptions, etc., are also centralized into the Health Fog. After collecting all data resources it is preprocessed and curated for activity detection and appear as intermediate data. This intermediate data are then shared with the doctor and the nutritionist or as preferred by the user. The calorie meter, activity detection, activity detail, and personalized recommendations are also shared with the users.

The data acquired from accelerometer and GPS followed by the detected activity against this data are shown in Fig. 3. Due to space limit the visibility of raw data has been restricted with fewer rows only followed by their transformation into useful information of activity on Health Fog. Depending upon the activity detection, the burned calories are calculated and logged. The final data for activity detection and calories are then deposited on the cloud and shared with the Hospital and nutritionist. The authorized entities can monitor the selected data as required. Figure 4 shows the total activities performed during a day to the user along with recommendations by the doctor and the nutritionist. The total amount of calory intake is shown as

Accelerometer Data

User Device ID	X-coordinate	Y-coordinate	Z-coordinate	timestamp
25	16	14	14	11/26/2014 8:16
25	16	14	15	11/27/2014 8:16
25	16	14	15	11/28/2014 8:16
25	16	14	17	11/29/2014 8:16
25	16	14	18	11/30/2014 8:16
25	16	14	21	12/1/2014 8:16
25	16	14	22	12/2/2014 8:16
25	16	14	24	12/3/2014 8:16
25	16	14	25	12/4/2014 8:16
25	16	14	27	12/5/2014 8:16
25	16	14	28	12/6/2014 8:16
25	16	14	29	12/7/2014 8:16
25	16	14	29	12/8/2014 8:16
25	16	14	32	12/9/2014 8:16

GPS Data

User Device ID	lattitude	longitude	speed	timestamp
25	12	16	16	11/26/2014 8:16
25	12	16	18	11/27/2014 8:16
25	12	16	18	11/28/2014 8:16
25	12	16	19	11/29/2014 8:16
25	12	16	19	11/30/2014 8:16
25	12	16	21	12/1/2014 8:16
25	12	16	22	12/2/2014 8:16
25	12	16	23	12/3/2014 8:16
25	12	16	23	12/4/2014 8:16
25	12	16	24	12/5/2014 8:16
25	12	16	25	12/6/2014 8:16
25	12	16	25	12/7/2014 8:16
25	12	16	26	12/8/2014 8:16
25	12	16	26	12/9/2014 8:16

Activity Recognition

User Device ID	Activity	Start Time	End Time
25	Cycling	11/26/2014 8:16	11/26/2014 8:20
25	Jogging	11/26/2014 8:35	11/26/2014 8:55
25	Running	11/26/2014 9:35	11/26/2014 9:40
25	Jogging	11/26/2014 10:00	11/26/2014 10:10
25	Walking	11/26/2014 10:20	11/26/2014 10:25
25	Walking	11/27/2014 9:35	11/27/2014 9:50
25	Cycling	11/27/2014 10:00	11/27/2014 10:20

Fig. 3 Raw data for accelerometer and GPS along with activity recognition data

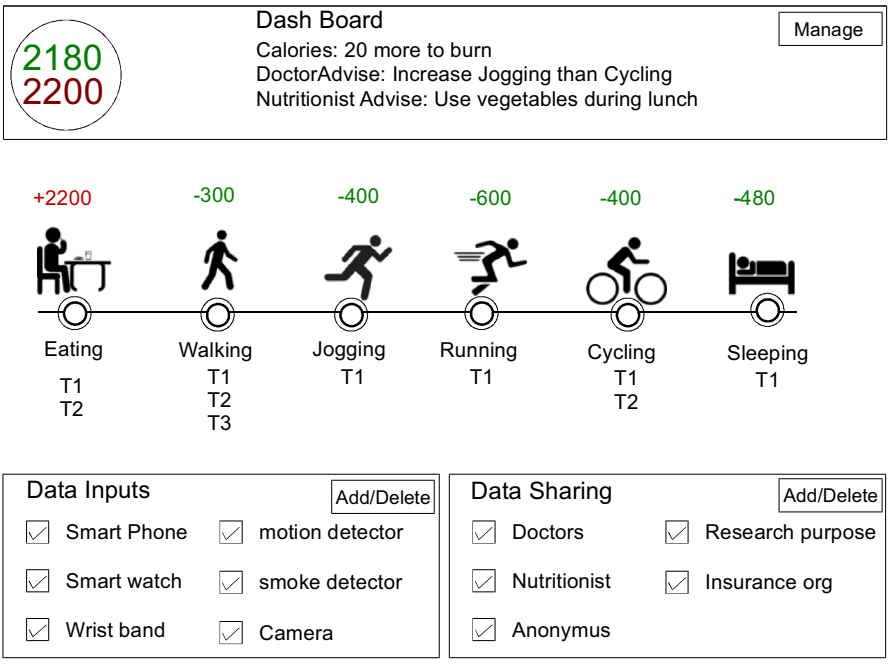


Fig. 4 Health Fog: user interaction

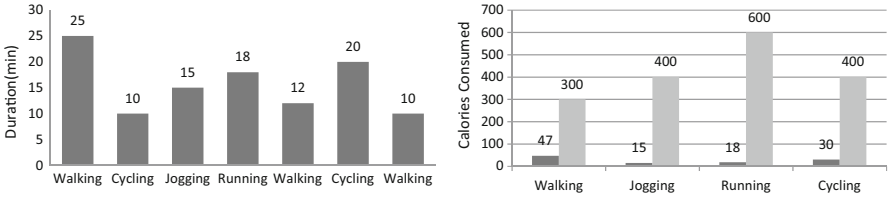


Fig. 5 Calories' consumption

2200, whereas calorie consumption by each activity is shown separately. The number of times an activity is performed during a day is shown from $(T_1, T_2, \ldots, T_n)$. The activities performed along with the food intake are then shared with the doctor on regular basis for appropriate advice and cure. The data input resources can be selected as per user's own preferences. Likewise, the control of sharing user's personal data (without disclosing her identity) with other entities is also under the discretion of user. The daily breakup of information comprising calories' break down, activity detection, duration, and recommendations by the doctors and nutritionist is shown to the user as appearing in Fig. 5.

The information which is uploaded from the Health Fog to the cloud is evaluated on local machine and Google App Engine [17] as cloud server. The local machine specification comprises on Intel(R) Core(TM) i3 processor and 4GB of RAM, whereas Microsoft(R) Windows7(TM) X 64bit is the OS installed.

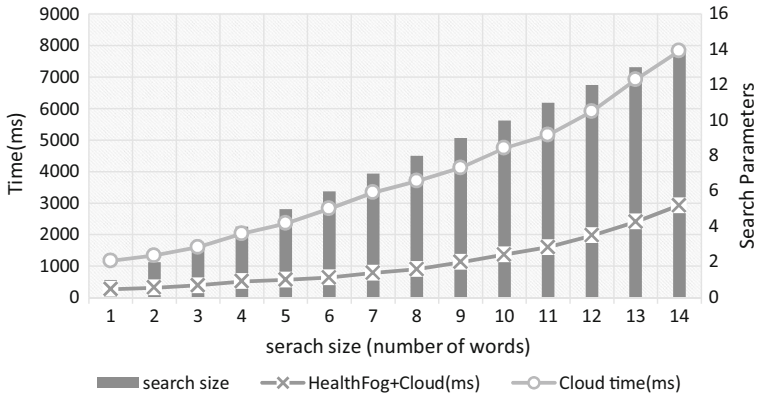


Fig. 6 Comparison: execution time

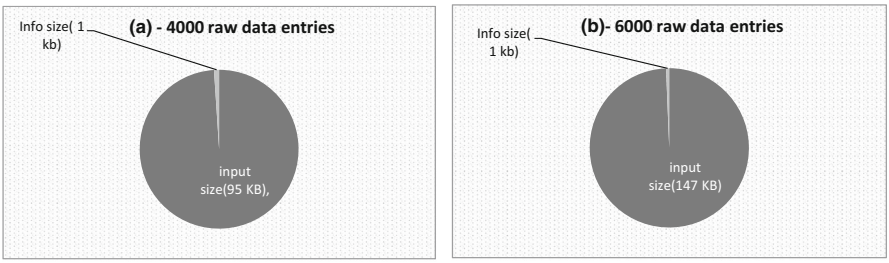


Fig. 7 Data reduction

The performance gain with respect to execution time on the information which is preprocessed through Health Fog and without its preprocessing is shown in Fig. 6. Here different selection parameters like user age, gender, BMI index have been used for the test purpose. This performance gain is achieved due to the transformation of raw data into required information thus eliminating unnecessary communication. The data reduction is shown in Fig. 7 where 4000 raw data entries occupying 95 Kb of space are replaced with only 1 Kb of actual information. Similarly, 147 Kb of space for 6000 Kb raw data entries is again transformed into 1 Kb of actual information thus saving notable communication overhead due to unnecessary movement of raw data.

7 Discussion

The ever-growing demand for ubiquitous healthcare systems to improve human health and well-being has suitably engaged advanced technologies, namely cloud computing, IoT, sensory devices, and wearables. With this adoption, the opportunity of including multivariate data in healthcare applications has been made possible for personalized and patient centric services. For real-time provisioning of data from this technology, their omnipresence has been made available through smart phones and wearables for continuous monitoring. With the development of internet of things (IoT) and smart

devices the data nodes are increasing at an exponential rate inviting big data to be stored and processed on cloud [18] due to certain reasons. These reasons primarily address the issues for central storage, complex computation, and information sharing. In an effort to optimize the whole process, Fog computing, which is an evolving paradigm shift, can facilitate such systems as a gateway between the end user and cloud. The constructive control of Fog computing effectively minimize the unnecessary communication from data generating nodes to cloud. In addition, certain policies and rules can be integrated within the Fog to ensure data privacy and security. In conventional applications that are solely dependent on cloud, the latency increases and required quality of services degrades. For this purpose, Fog computing as an intermediary layer between the cloud and end user plays its pivotal role for low latency, better visualization, and context awareness [8,19]. Fog computing can also increase the security in the public cloud. A trustworthy cloud provider is necessary but accidents still tend to happen and information gets lost. The exposure risk for information leakage can be limited which is sent to the cloud by initial processing on the fog [20]. Complex security challenges are being faced by big data and cloud computing. The most promising way to deal with these challenges is fog computing [21].

In recent years Cloud and Fog computing have drawn profound attention in e-health related systems and applications. In [24], authors present a cloud computing solution for patient data collection in health care institutions. The proposed system uses sensors attached to medical equipment to collect patient data and sends the data to cloud for providing ubiquitous access. Introducing smart gateways Chen et al. [25] introduce a smart gateway for health care system using wireless sensor network. The proposed gateway acting as a bridge between wireless sensor network and public communication networks has a data decision system. In [26], authors propose a mobile gateway for ubiquitous health care system using ZigBee and Bluetooth. The gateway presents various services such as alarms and analysis of medical data. Yang et al. present a personal health monitoring gateway based on smartphone [27]. The proposed gateway uses a Bluetooth interface to upload gathered data to remote servers. In [28] mobile fog is introduced for future internet applications which will be geographically distributed and are latency sensitive. Mobile fog consists of different devices like smartphones, smart watches, tablets, and even drones. A case study given by Tuan et al. [29] highlights the feasibility of IoT to monitor human health in real-time using ubiquitous health monitoring systems. Their proposed health monitoring system exploit the concept of Fog computing at smart gateways providing advanced techniques and services such as embedded data mining, distributed storage, and notification service at the edge of network.

8 Conclusion and future work

In this paper we have proposed a framework of Health Fog for sharing and processing health-related information based on data acquired from multiple resources. We have used Fog computing features as an intermediary layer between the cloud server and end user to avoid unnecessary flow of information and better control over data privacy and

security while processing and sharing the information. In future extension to Health Fog, we will incorporate the social media as an input data resource and also we will build the knowledge reservoir accrued over single instances of Health Fog users. With this addition the feature scale of Health Fog will be equipped with more usability and shareability.

Acknowledgments This work was supported by a grant from the Kyung Hee University in 2013[KHU-20130439].

References

1. FitnessKeeper (2015) Runkeeper everyone every run. http://runkeeper.com/
2. Noom (2015) Noom help you to make a change. https://www.noom.com/
3. Azumio (2015) Argus quantify your day-to-day. http://www.azumio.com/s/argus/index.html
4. myfitnesspal (2015) Lose weight with myfitnesspal. https://www.myfitnesspal.com/
5. Fitocracy (2015) Fitocracy. https://www.fitocracy.com/
6. Digifit (2015) Digifit personal engagement solution. http://www.digifit.com/personal-solutions.html
7. MIOLink (2015) Mio train with heart. http://www.mioglobal.com
8. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, pp 13–16
9. Firstbrook P, MacDonald N (2015) The growing importance of cloud access security brokers. https://www.gartner.com/doc/2032015/growing-importance-cloud-access-security
10. HP cloud access security protection platform. Business white paper : Case for a cloud access security brokers. hp.com/go/CloudAccessSecurity, (2015)
11. Fischlin M (2001) A cost-effective pay-per-multiplication comparison method for millionaires. In: Topics in Cryptology CT-RSA 2001, Springer. pp 457–471
12. Sweeney Latanya (2002) Achieving k-anonymity privacy protection using generalization and suppression. Int J Uncertain Fuzziness Knowl Based Syst 10(05):571–588
13. Sweeney Latanya (2002) k-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl Based Syst 10(05):557–570
14. Evdokimov S, Günther O (2007) Encryption techniques for secure database outsourcing. In: Computer Security–ESORICS 2007, Springer, pp 327–342
15. De Capitani S, Di Vimercati FS, Jajodia S, Paraboschi S, Samarati P (2007) A data outsourcing architecture combining cryptography and access control. In: Proceedings of the 2007 ACM workshop on Computer security architecture, ACM, pp 63–69
16. Curino C, Jones EPC, Popa RA, Malviya N, Wu E, Madden S, Balakrishnan H, Zeldovich N (2011) Relational cloud: a database-as-a-service for the cloud
17. Google (2013) Google app engine. https://cloud.google.com/products/app-engine
18. Shi Y, Ding G, Wang H, Roman HE, Lu S (2015) The fog computing service for healthcare. In: 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), IEEE, pp 1–5
19. Stojmenovic I, Wen S (2014) The fog computing paradigm: Scenarios and security issues. In: 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, pp 1–8
20. Stolfo SJ, Salem MB, Keromytis AD (2012) Fog computing: mitigating insider data theft attacks in the cloud. In: 2012 IEEE Symposium on Security and Privacy Workshops (SPW), IEEE, pp 125–128
21. Yannuzzi M, Milito R, Serral-Gracia R, Montero D, Nemirovsky M (2014) Key ingredients in an iot recipe: fog computing, cloud computing, and more fog computing. In: 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp 325–329
22. Paillier P (2000) Trapdooring discrete logarithms on elliptic curves over rings. In: Okamoto T (ed) Advances in CryptologyASI—ACRYPT 2000. Springer, Berlin, Heidelberg, pp 573–584
23. Paillier P (1999) Public key cryptosystems based on composite degree residuosity classes. In: Proceedings of the 17th international conference on Theory and application of cryptographic techniques, EUROCRYPT'99, Springer-Verlag, Berlin, Heidelberg, pp 223–238

24. Rolim CO, Koch FL, Westphall CB, Werner J, Fracalossi A, Salvador GS (2010) A cloud computing solution for patient's data collection in health care institutions. In: Second International Conference on eHealth, Telemedicine, and Social Medicine, 2010. ETELEMED'10, IEEE, pp. 95–99
25. Chen Y, Shen W, Huo H, Xu Y (2010) A smart gateway for health care system using wireless sensor network. In: 2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM), IEEE, pp 545–550
26. Laine TH, Lee C, Suk H (2014) Mobile gateway for ubiquitous health care system using zigbee and bluetooth. In: 014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2. IEEE, pp 139–145
27. Yang S, Gerla M (2011) Personal gateway in mobile health monitoring. In: 011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2. IEEE, pp 636–641
28. Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B (2013) Mobile fog: a programming model for large-scale applications on the internet of things. In: Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing. ACM, pp 15–20
29. Gia TN, Rahmani MJAM, Westerlund T, Liljeberg P, Tenhunen H (2015) Fog computing in healthcare internet-of-things: a case study on ecg feature extraction
30. Hussain J, Khan WA, Afzal M, Hussain M, Kang BH, Lee S (2014) Adaptive user interface and user experience based authoring tool for recommendation systems. In: Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services. Springer, pp 136–142