IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 3, 2021, accepted July 26, 2021, date of publication July 30, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101308

Secure Health Fog: A Novel Framework for Personalized Recommendations Based on Adaptive Model Tuning

UBAID UR REHMAN[1,2], SEONG-BAE PARK[1], (Member, IEEE), AND SUNGYOUNG LEE[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, Kyung Hee University (Global Campus), Giheung-gu, Yongin-si, Gyeonggi-do 17104, South Korea
[2]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan

Corresponding authors: Seong-Bae Park (sbpark71@khu.ac.kr) and Sungyoung Lee (sylee@oslab.khu.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program Supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2017-0-01629, in part by IITP Grant by the Korean Government through MSIT under Grant 2017-0-00655, and in part by the MSIT, South Korea, through the Grand Information Technology Research Center Support Program Supervised by the IITP under Grant IITP-2021-2020-0-01489 and Grant NRF-2019R1A2C2090504.

ABSTRACT The emergence of smart technology has equipped humans with wearables and sensors that collect relevant data related to individuals and their surroundings. In healthcare, the collected data can monitor user emotion, behavior, and activity, which leads to the development of personalized decisions and improves lifestyle. In this paper, we analyzed the existing health fog framework and identified its limitations in terms of security, performance, and accuracy. Based on these limitations, we propose a secure health fog (SHF) framework that collects data from different Internet of Things (IoT) devices and maintains a personalized repository for adaptive model tuning. The adaptive model improves periodically based on user feedback and generates a personalized recommendation. Moreover, the existing IoT devices mostly rely on low-cost and low-power Zigbee technology, which is vulnerable to different attacks, such as device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay attack. Therefore, we propose a Zigbee Secure Health Fog (ZigbeeSHF) protocol, which uses symmetric and public-key cryptography to prevent these attacks. For data migration security, we concatenate the encrypted data with the encrypted digital signature to provide data authenticity, integrity, and confidentiality. To support our claims, we use the automated formal verification tools Scyther and AVISPA (Automated Validation of Internet Security Protocols and Applications), which evaluate the protocols based on the threat model and exploits the vulnerabilities in different attacking environments. The results of both tools ensure prevention against the mentioned attacks. As a proof of concept, we also evaluate the accuracy and performance of our proposed framework in a smart studio apartment. The result shows that the adaptive model tuned for an individual user is very effective, and the average accuracy of 32.5% is improved after one month. Furthermore, the proposed ZigbeeSHF protocol requires 7.93% and 25.35% more computation time than Zigbee with and without installation code, respectively.

INDEX TERMS Secure health fog, adaptive model tuning, personalized decision, security, Zigbee.

I. INTRODUCTION

With the emergence of smart technology and ubiquitous computing, different sensors and wearable devices can collect information related to a specific user and their surrounding environment. The information collected from these sensors can facilitate the user in their daily life, such as task reminder, medication intake, fitness, adjustment of room temperature, lighting, ventilation, etc. Such innovation makes the user's life manageable and improves lifestyle. These innovations depend on activity recognition, which is currently a mature and popular area of research. The literature related to activity

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhui Yuan.

recognition has focused on activity monitoring [1], machine learning modelling [2], sensor technology [3], behavior tracking [4], and emotion recognition [5]. Several healthcare applications have also been proposed, including smart clothing [6], health trackers for weightlifters [7], patient fall detection [8], sedentary behavior identification [9], and recommendations for chronic disease patients [10]. These applications are linked with specialized medical devices or embedded sensors, which continuously monitor patient activities (such as muscle activity, heart rate, calories burned, and sedentary behavior) and provide recommendations accordingly.

The importance of such research is to support communities, especially the aging population, to live safely at home. The healthcare sector faces severe challenges in terms of quality-of-life, hospital capacity, and nursing care longevity. According to global demographic trends [11], the world population is aging due to the increase in life expectancy and decrease in childbirth, especially in high-income countries such as the Republic of Korea, Liechtenstein, Sweden, and Switzerland. The government is taking necessary steps to overcome these problems by providing training and monetary incentives for family members to take care of elderly persons. With increasing age, the older population suffers from different chronic diseases and thus requires continuous monitoring. However, it is challenging for working family members to provide intensive care to the elderly, especially those with cognitive impairment, physical disabilities, or dementia, which require care around the clock. Therefore, family members hire a nurse or send their elderly to nursing care homes, which has a negative impact on elderly health in terms of depression and anxiety.

The research community proposed different solutions using IoT devices, wearable sensors, and smartphones to monitor the patient's health conditions and provide relevant healthcare services at home. These devices generate a large amount of data and require computational power for performing tasks such as the analysis, aggregation, storage, and generation of context-sensitive decisions. The healthcare application deals with personally identifiable information that belongs to patients. Therefore, the traditional sensor-to-cloud architecture is not preferred due to privacy concerns, network failures, and data transmission latency issues that may put a patient's life at stake. In healthcare, efficiency and reliability play an important role in real-time operations. Thus, the traditional sensor-to-cloud architecture was extended to a decentralized computing infrastructure called fog computing, which collects the data from underlying IoT devices, processes resource-hungry or sensitive operations at the network edge and sends relevant data to the cloud for long-term storage. Fog computing acts as an intermediate layer between cloud and end devices. Similar to cloud computing, fog also delivers different services in terms of health fog [12], vehicular fog [13], and energy fog [14].

In this paper, we focus on the health fog that delivers healthcare-as-a-fog service and provides efficient services to the patient regarding monitoring, medication adherence, and behavior tracking. According to our analysis, the existing literature related to health fog has primarily focused on performance measures and considered security as a secondary feature due to computational overhead, which leads to several cyber-attacks, such as device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay. To prevent these attacks, we propose a secure health fog (SHF) framework that acts as an intermediate layer between underlying IoT devices and the cloud. The SHF securely collects the data from underlying IoT devices, analyzes the data, and maintains a personalized repository for adaptive model tuning, supporting context-sensitive decision making. Moreover, due to the sensitive nature of health data, our proposed framework provides data ownership, where the user can select the data to collect/discard during decision making and data migration. Furthermore, we propose security protocols to ensure prevention against the mentioned attacks. The main contributions of this study are described as follows:

- We propose a secure health fog framework that considers security, performance, and accuracy as primary factors.
- The secure health fog maintains a personalized repository based on user feedback that evolves the model and supports personalized, context-sensitive decision making.
- We describe the Zigbee Secure Health Fog (ZigbeeSHF) protocol, which considers the limitations of the existing Zigbee technology and prevents different attacks, such as device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay.
- We present the data migration protocol, which concatenate the encrypted data with the encrypted digital signature to provide data authenticity, integrity, and confidentiality.
- We evaluate the protocols using the formal verification tools Scyther and AVISPA, which evaluate the protocol based on the threat model and exploit vulnerabilities in different attacking environments.
- We deploy the secure health fog in a smart studio apartment to evaluate the effect of adaptive model tuning on personalized context-sensitive decision making. Additionally, the performance of the proposed protocols is evaluated in terms of processing time and CPU utilization.

The rest of this paper is organized as follows. Section II provides an overview of relevant studies. Section III briefly describes the methodology of SHF framework. The evaluation of our proposed methodology is presented in section IV. Section V discusses the detailed analysis of results. Finally, section VI concludes the proposed work and sets future directions.

II. RELATED WORK

In this section, we have described the literature that provides healthcare as a fog service. Fog computing is considered an

extension of cloud computing and is relatively more secure due to the data being analyzed and maintained on a local fog node closer to the data source. However, many challenges related to security and privacy exist when the fog node shares data with other resources to perform a specific task. Similar to the situation in other domains, personalized analysis and recommendation has become a necessity of healthcare. Therefore, in this section, we present the most relevant research studies that propose the health fog solution and support a healthy lifestyle. In our survey of the literature, we found only nineteen research studies that provide healthcare as a fog service. We classified these studies into the Health Fog Methodology and Health Fog Algorithm, which include thirteen and six research studies, respectively. The comprehensive description of the selected studies is described as follows:

A. HEALTH FOG METHODOLOGY

The health fog methodology considers the process in which healthcare application delivers a service to the end user through fog computing. In [15], the authors proposed a three-layered architectural model for smart health infrastructure, which measures the patient vital signs regularly and helps in quality treatment. Fratu *et al.* described the concept of monitoring pulmonary disease and mild dementia patients in [16], which analyzes the patient medical condition and generates a recommendation regarding the actions that need to be taken. The study considered relevant performance measures and suggested the use of fog to reduce latency. Similarly, the benefits of fog computing in chronic obstructive pulmonary disease were described in [17] to monitor patient oxygen dosage, energy consumption, and activity. Then, the information was processed using the fog to generate a decision regarding the oxygen dosage. The proposed system provides efficient processing mechanisms, which help patients perform some physical activity. In [18], a fog computing interface was proposed that processed clinical speech data from patients with Parkinson's disease and identifies features such as perceptual loudness, short-term energy, zero-crossing rate, and spectral centroid to provide an assessment of the patient's condition. The authors used a secure copy protocol to ensure communication security. Moreover, a service-oriented fog computing architecture named fog data was proposed in [19], which collected the data from wearable sensors and stores the recurring patterns along with clinical information on the cloud. The preliminary analysis and filtering were performed using fog computing services that resulted in data reduction, low power consumption, and high efficiency. Gia *et al.* designed a healthcare monitoring system for cardiac disease using fog computing [20], which extracts features from electrocardiograms and uses them to diagnose cardiac diseases. The results show that with the usage of fog computing, high efficiency and low latency can be achieved. Furthermore, a system architecture for augmented brain-computer interfaces based on fog computing and linked data was proposed in [21], which uses ubiquitous computing services, wireless EEG headsets, and smartphones to detect the user's brain activity. The fog server in this study acts as a data broker that collects the data from the EEG sensors and publishes it as linked data with low latency. In [22], the authors described the strategic position of gateway in a fog computing environment to monitor patient health remotely. The system was deployed and tested for patients with acute illness, but it did not describe the accuracy or interagreement level of the proposed solution. Farahani *et al.* [23] conducted a survey and designed an IoT-based eHealth ecosystem that considered traditional healthcare system challenges such as data management, scalability, interoperability, standardization, regulatory affairs, and security. In [24], the authors deployed a secure centralized fog computing architecture, which securely collects the data from sensors and after processing sent to the cloud for seamless access. The study emphasized the location-based tracking of authorized devices, but it did not focus on the effect of security on performance and accuracy. Hassen *et al.* proposed a home hospitalization system that uses sensing units (such as vital signs and environmental factors) and supports users in monitoring their vitals on smartphones [25]. Tuli *et al.* integrated ensemble deep learning in edge computing devices and designed a health fog framework [26], which efficiently manages heart patient data and delivers healthcare services. In [27], the authors designed and developed a fog-assisted health monitoring system, which analyzes the biological signs of the patient and informs the medical specialist in the case of abnormality detection.

B. HEALTH FOG ALGORITHM

The health fog algorithm describes the set of rules according to which a healthcare application solves a specific problem and delivers it to the end user using fog computing. Cao *et al.* [28] proposed a fall detection algorithm based on acceleration magnitude values, nonlinear time series analyses, and filtering techniques that detect falls in the case of a stroke. The proposed algorithm was evaluated with real-world data, which achieved high sensitivity and high specificity, with a minimum response time. Similarly, an e-health system was proposed in [29], which detects falls and gas leakage in a smart environment. The proposed algorithm informs the caretaker in case of an emergency and supports location awareness. In [30], the authors designed a medical cyber-physical system that uses fog computing for task distribution and provides a cost-efficient solution. The proposed algorithm was evaluated based on performance and cost-effectiveness. Vora *et al.* designed TILAA [31], a tactile-based framework that analyzes patient health conditions using multiple sensing devices and sends tactile feedback to the healthcare service provider in case of an emergency. Similarly, the authors of [32] proposed a fog-assisted continuous monitoring system, which remotely analyzes patient health conditions (such as glucose level, ECG, falls, and activities) and provides advanced services (such as interoperability, security, and local data storage). The proposed approach considered the performance measure and encrypted

the data at the fog node. In [33], the authors proposed an intelligent system that analyzes the environmental condition using IoT devices and alerts the individual infected with the dengue virus. The proposed system considered accuracy, security, and performance measures, but symmetric key encryption was used to provide data source authentication and was vulnerable to replay attacks.

C. ZIGBEE TECHNOLOGY

Recently Zigbee has been deployed in many smart devices due to its low cost, low power, and low complexity [34]. In this section, we summarize the literature related to Zigbee technology that exploits vulnerabilities and propose a security solution. The initial version of Zigbee was proposed in 2004 [35]; since then, many security enhancements have been implemented to provide secure communication [36]. Currently, IoT devices rely on Zigbee 3.0, which supports an optional feature of installation code [37]. The installation code is a quick response (QR) code affiliated with each device and supports deriving the link key using Advanced Encryption Standard-Matyas-Meyer-Oseas (AES-MMO). The derived link key is used to encrypt the network key and ensure data authenticity [38]. In Zigbee 3.0, the attacker mainly exploits the association phase and launches different attacks, such as communication disruption, fake device injection, privacy leakage, and device control [39]. Wang *et al.* proposed a protocol that uses an elliptic-curve Diffie Hellman (ECDH) along with the installation code and ensures prevention against the mentioned attacks [39]. The security of ECDHs depends on the pre-shared secret, and exploitation can compromise communication security [40]. Okada *et al.* used indirect transmission in Zigbee and launched a low-rate denial of service attack that eluded the preventive measures [41]. In [42], the author proposed a certificate-less key agreement protocol that uses elliptic curve cryptography and prevents impersonating attacks.

D. ANALYSIS OF LITERATURE SURVEY

In this study, we focused on the health fog, where the recommendation needs to be accurate, with no or minimum latency, and contain high-security measures. Any error, delay, or tampering may lead to a life-threatening situation. Therefore, we critically analyzed these studies in terms of security, performance, and accuracy, as shown in Table 1. According to our analysis, sixteen studies ([16]–[23], [26]–[33]) have focused on improving the performance of their proposed solution using fog computing, which justifies the observation that fog computing improves efficiency and performance. However, limited studies have considered security ([18], [22], [24], [27], [32], [33]) and accuracy ([26], [28], [29], [31], [33]). Theoretically, accuracy is linked with decision-making and recommendation, but it needs to be considered an important health fog factor. Most of the existing studies considered security as a secondary feature due to its computational overhead and negative correlation with performance. Based on these limitations, we have proposed a secure health fog framework that considers security, performance, and accuracy as primary factors.

TABLE 1. Summary of literature review (considered(✓), not-considered(✗)).

SN	Research Studies	Security	Performance	Accuracy
1	Stantchev et al. [15]	✗	✗	✗
2	Fratu et al. [16]	✗	✓	✗
3	Masip et al. [17]	✗	✓	✗
4	Monteiro et al. [18]	✓	✓	✗
5	Dubey et al. [19]	✗	✓	✗
6	Gia et al. [20]	✗	✓	✗
7	Zao et al. [21]	✗	✓	✗
8	Rahmani et al. [22]	✓	✓	✗
9	Farahani et al. [23]	✗	✓	✗
10	Vilela et al. [27]	✓	✓	✗
11	Hassen et al. [25]	✗	✗	✗
12	Tuli et al. [26]	✗	✓	✓
13	Thota et al. [24]	✓	✗	✗
14	Cao et al. [28]	✗	✓	✓
15	Gu et al. [30]	✗	✓	✗
16	Gia et al. [32]	✓	✓	✗
17	Vora et al. [31]	✗	✓	✓
18	Craciunescu et al. [29]	✗	✓	✓
19	Sood et al. [33]	✓	✓	✓
20	Proposed Framework	✓	✓	✓

Moreover, the sensors deployed in our smart studio apartment belong to the Zigbee family (CC1352R [43], CC1352P [44]). Therefore, we have analyzed the existing literature related to Zigbee and identified that the protocol uses weak authentication in terms of installation code during the association phase [38], which makes the protocol vulnerable to a variety of attacks [39]. Researchers have proposed different approaches to ensure prevention against these attacks [35], [40], [42], but the proposed solution opens the door for new variants of cyber-attacks [41]. Therefore, we have proposed the ZigbeeSHF protocol, which considers the limitations of existing studies and provides prevention against different attacks, such as device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay.

III. PROPOSED SECURE HEALTH FOG FRAMEWORK

As discussed in the section II, a limited number of studies have focused on accuracy and security aspects, which need to be considered along with the performance measure to gain user trust. It may be possible that a malicious fog device becomes a part of the network and performs different attacks, such as identity theft, tempering, and ransomware. Therefore, we considered all these aspects and propose a secure health fog framework that overcomes the limitations of the existing literature and maintains a personalized repository for adaptive model tuning. Figure 1 presents the layered architecture of our proposed framework. The description of each layer is described as follows. Moreover, a list of frequently used symbols and abbreviations is given in Table 2.

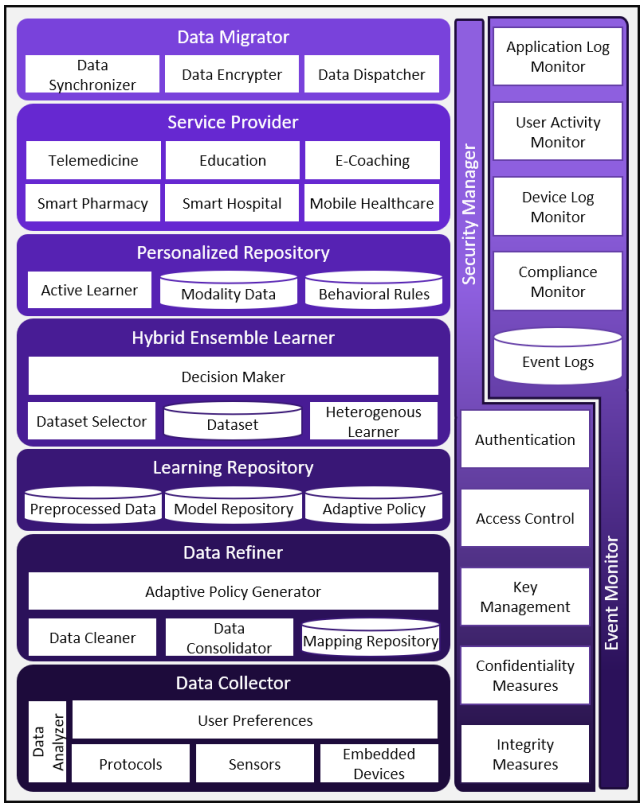

FIGURE 1. Layered architecture of Secure Health Fog framework.

Algorithm 1 Data Collector Layer

Input: d_{mtx}, c_{ommP}, u_{pref}
Output: d_{mtxls}, c_{ommPls}, $u_{prefBehr}$
/* scanning state & detect IoT devices */
while *true* **do**
  if *device* $\vDash$ *conditions* **then**
    $d_{mtxls} \leftarrow d_{mtx}$
    $c_{ommPls} \leftarrow c_{ommP}$
split u_{pref} into c_{ond} and c_{oncl}
$c_{ond} \leftarrow$ infer and map h_{lc} for c_{ond}
$c_{oncl} \leftarrow$ infer and map h_{lc} for c_{oncl}
$u_{prefBehr} \leftarrow c_{ond} \,||\, c_{oncl}$

A. DATA COLLECTOR

The data collector connects via a gateway to the underlying IoT devices and collects the raw data, which includes the communication protocol (c_{ommP}), data modality matrix (d_{mtx}), and user preference (u_{pref}). Algorithm 1 describes our proposed approach for the data collector layer. Initially, the scanning state is activated to identify the list of available IoT devices over the network. To become a part of the health fog, each detected device needs to fulfill certain conditions, such as authentication, authorization, and access control, that prevent intruders and malicious devices. The data collector collects two types of information for each authorized device: (i) generated data in the form of a modality matrix (the x, y, and z coordinates of the accelerometer are considered a modality matrix); and (ii) communication protocols such as TCP/UDP that help us ensure transport layer security. Then, user preferences (u_{pref}) that reflect the nature or behavior of an individual (such as if sitting in the living room then adjust temperature) are acquired. The data analyzer splits u_{pref} into condition (c_{ond}) (sitting in the living room) and conclusion (c_{oncl}) (adjust temperature). Then, the identified high-level context (h_{lc}) (activity, location, and action) is mapped using a rule-based approach to generate a rule, which concatenates the condition (c_{ond}) and conclusion (c_{oncl}), such as IF (activity=sitting, location=living room) THEN (action=adjust temperature).

Algorithm 2 Data Refiner Layer

Input: d_{mtxls}, c_{ommPls}, $u_{prefBehr}$
Output: d_{mtxMap}, $u_{prefBehr}$, l_{hRule}
$\forall d_{mtxls}$
if *missing value exist* **then**
  Impute missing value with the column means
  $d_{mtxls} \leftarrow$ update d_{mtxls}
else if *duplicated value exist* **then**
  remove duplicated rows
  $d_{mtxls} \leftarrow$ update d_{mtxls}
else
  do nothing
foreach h_{lc} *in* $u_{prefBehr}[c_{ond}]$ **do**
  $l_{lc} \leftarrow$ identify l_{lc} from M_{apRepo}
  if l_{lc} *exists in* d_{mtxls} **then**
    $d_{mtx} \leftarrow$ extract d_{mtx} from d_{mtxls}
    $c_{ommP} \leftarrow$ extract c_{ommP} from c_{ommPls}
    $c_{ond} \leftarrow l_{lc}$
    $c_{oncl} \leftarrow h_{lc}$
  $d_{mtxMap} \leftarrow$ map $c_{oncl}||c_{ond}||d_{mtx}||c_{ommP}$
  $l_{hRule} \leftarrow c_{ond}||c_{oncl}$

B. DATA REFINER

The data refiner processes the raw data, removes inconsistencies and generates adaptive policies for personalized recommendations. Algorithm 2 describes our proposed approach for the data refiner layer. It takes lists of data modality matrices (d_{mtxls}), communication protocols (c_{ommPls}), and relations between modalities and user preferences ($u_{prefBehr}$) as inputs. The data cleaner analyzes the d_{mtxls}, checks for missing and duplicated values, imputes the column mean value and removes duplicated rows. The data consolidator extracts the high-level context (h_{lc}) from the $u_{prefBehr}$ condition (c_{ond}) (such as activity and location), and then identifies the low-level context (l_{lc}) for each identified h_{lc} using the mapping repository (M_{apRepo}). M_{apRepo} contains a predefined list of modalities mapped from h_{lc} to l_{lc} (such as activity (h_{lc})

TABLE 2. Nomenclature of Secure Health Fog framework.

Abbreviation	Meaning	Abbreviation	Meaning
$t_{\hat{rSet}}$	Output of base learner as training set	l_{lc}	Low level context
$\models$	Satisfy	M_{apRepo}	Mapping repository
a_{ctrl}	Access control	m_{Attp}	Matched attributes in predicted matrices
a_{nnData}	Annotated data	m_{Attr}	Matched attributes in rule
a_{pRepo}	Adaptive policy repository	M_{feat}	Meta-features
a_{uth}	Authentication	m_{igrT}	Migration time interval
a_{uthAtt}	Authentication attempt	M_{model}	Meta-model
$a_{uthFlag}$	Authentication flag	$m_{odelRepo}$	Model repository
b_{ehRepo}	Behavioral rule repository	map	Map and assign values
b_{sLAlgo}	Base learning algorithm	p	Predicted value
c_{evt}	Current event data	p_{ls}	Predicted matrices list
c_{ommP}	Communication protocol	p_{pRepo}	Preprocessed data repository
c_{ommPls}	List of communication protocol	P_{Repo}	Personalized modality data repository
c_{oncl}	Conclusion	p_{tmodel}	Pre-trained model
c_{ond}	Condition	p_k	Public key
c_{onfl}	Conflicts	q_{ls}	List of queries
$c_{onfMeas}$	Confidentiality measure	r	Rule
$c_{ryptoHash}$	Cryptographic hash function	r_{ec}	Recommendation
c_{xID}	Consumer identity	r_{eg}	Registration
d_{ataSet}	Dataset	r_{es}	Response
$d_{isprFlag}$	Dispatcher flag	r_{esls}	List of responses
d_{migrF}	Data migration flag	r_{ls}	List of rules
d_{mtx}	Device modality matrix	$S_{ecSvReq}$	Security service request
d_{mtxls}	List of device modality matrix	$S_{ecSvRes}$	Security service response
d_{mtxMap}	Mapped device modality matrix	S_{essID}	Session identifier
d_{retrv}	Retrieved data	S_{fogN}	Source fog node
D_{sig}	Digital signature	S_{vReq}	Service request
d_{sRepo}	Dataset repository	S_{vRes}	Service response
e_{alert}	Event alert	S_{ync}	Synchronization
$E_{D_{sig}}$	Encrypted digital signature	s_k	Private key
e_{data}	Event data	t	Time
$e_{logRepo}$	Event log repository	T_{cloud}	Targeted cloud platform
E_{ncrypt}	Encryption	t_{hld}	Threshold
E_d	Encrypted data	t_{rSet}	Training set
f_{NRepo}	Fog node repository	t_{tlAttr}	Total attributes in rule
h_{lc}	High level context	$t_{tlAttrp}$	Total attributes in predicted matrices
I_{meas}	Integrity measure	t_{tSet}	Testing set
infer	Request relevant function for solution	TTP	Trusted third party
k	Symmetric key	u_{pref}	User preference
K_{gen}	Cryptographic key generator	$u_{prefBehr}$	User preferences behavior rules
K_{mgt}	Key management	u_{srFB}	User feedback
l_{hRule}	Low to high level rules	w_t	Weight

mapped to l_{lc} that includes an accelerometer and a gyroscope). Furthermore, the device modality matrix (d_{mtx}), communication protocol (c_{ommP}), l_{lc}, and h_{lc} are mapped based on the identified l_{lc} in d_{mtxls}, which describes the complete structure of the modalities. The adaptive policy generator concatenates the identified l_{lc} with h_{lc} and generates a rule structure such as c_{ond}=accelerometer, gyroscope and c_{oncl}=activity.

C. LEARNING REPOSITORY

The learning repository collects, stores, manages, and shares the data. This layer consists of three repositories: preprocessed data (p_{pRepo}), model repository ($m_{odelRepo}$), and adaptive policy (a_{pRepo}). Algorithm 3 describes our proposed approach for learning repositories. p_{pRepo} maintains refined data (d_{mtxMap}), which contains h_{lc}, l_{lc}, d_{mtx}, and c_{ommP}, such as activity || accelerometer || coordinates(x,y,z) || UDP. The model repository ($m_{odelRepo}$) loads the pretrained model (p_{tmodel}), such as predicting activity based on accelerometers and gyroscopes. a_{pRepo} maintains $u_{prefBehr}$ and l_{hRule}, which support personalized decision making. In the case of any query (q) or a list of queries (q_{ls}), this layer processes the

Algorithm 3 Learning Repository Layer

Input: d_{mtxMap}, $u_{prefBehr}$, l_{hRule}, q_{ls}
Output: r_{esls}

```
store d_mtxMap in p_pRepo
store u_prefBehr and l_hRule in a_pRepo
load p_tmodel in m_odelRepo
foreach query q in q_ls do
    r_es ← search q in p_pRepo, m_odelRepo, a_pRepo
    if r_es != null then
        r_es ← infer r_es
        r_esls ← add r_es
```

Algorithm 4 Hybrid Ensemble Learner Layer

Input: l_{hRule}, p_{tmodel}, $u_{prefBehr}$, d_{mtxMap}
Output: r_{ec}

foreach $l_{hRule}[c_{oncl}]$ *in* l_{hRule} **do**
  if p_{tmodel} *not exist* **then**
    load a_{nnData} in d_{sRepo}
    $d_{ataSet} \leftarrow$ select d_{ataSet} from d_{sRepo}, where $d_{ataSet} \equiv l_{hRule}$
    split d_{ataSet} in $\{t_{rSet\,i}, t_{tSet\,i}\}_{i=1}^{n}$
    for *i=1 to n* **do**
      $M_{feat\,i} \leftarrow b_{sLAlgo\,i}$ on $\{t_{rSet\,i}, t_{tSet\,i}\}$
    where $M_{feat\,i} = \{\hat{t_{rSet\,i}}, t_{tSet\,i}\}$ and $\hat{t_{rSet\,i}} = b_{sLAlgo}(t_{rSet\,1}), b_{sLAlgo}(t_{rSet\,2}), \ldots, b_{sLAlgo}(t_{rSet\,n})$
    $p_{tmodel} \leftarrow$ learn M_{model} on $M_{feat\,i}$
    store p_{tmodel} in $m_{odelRepo}$
  retrieve p_{tmodel} from $m_{odelRepo}$
  $d_{mtx} \leftarrow$ select $d_{mtxMap}[d_{mtx}]$, where $d_{mtxMap}[c_{oncl}] \equiv l_{hRule}[c_{oncl}]$ and $d_{mtxMap}[c_{ond}] \equiv l_{hRule}[c_{ond}]$
  p $\leftarrow$ predict p for d_{mtx} using p_{tmodel}
  $h_{lc} \leftarrow l_{hRule}[c_{oncl}]$
  $l_{lc} \leftarrow l_{hRule}[c_{ond}]$
  $p_{ls} \leftarrow$ map $h_{lc}||p||l_{lc}||d_{mtx}$
$\forall p_{ls}[p]$, **if** $p_{ls}[p]$ *exists in* $u_{prefBehr}[c_{ond}][h_{lc}]$ **then**
  get rule r from $u_{prefBehr}$
  $w_t = \frac{\sum_{i=0}^{n}(m_{Attr\,i}+m_{Attp_i})}{m_{Attr\,t}+m_{Attp_t}} \times 100$
  $r_{ls} \leftarrow r||w_t$
sort r_{ls} with w_t in descending order
if *r in* $r_{ls} \models$ *context* **then**
  $r_{ec} \leftarrow$ select top r
  $p_{ls} \leftarrow$ filter p_{ls} based on r_{ec}

query and returns the required information that includes data, model, and adaptive policies.

D. HYBRID ENSEMBLE LEARNER

The hybrid ensemble learner uses the combination of a heterogeneous machine learning algorithm and a rule-based approach to generate an appropriate recommendation after analyzing the input data modality matrix. Algorithm 4 presents the hybrid ensemble learner approach, which takes l_{hRule}, p_{tmodel}, $u_{prefBehr}$, and d_{mtxMap} as inputs and generates a recommendation. This layer analyzes the p_{tmodel} for each h_{lc} extracted from $l_{hRule}[c_{oncl}]$. If p_{tmodel} does not exist, then the annotated data (a_{nnData}) are loaded to the dataset repository (d_{sRepo}). The dataset selector selects the most appropriate data that contained similar features for predicting a specific label, such as accelerometer and gyroscope for activity prediction. The selected dataset (d_{ataSet}) then splits into n^{th} training (t_{rSet}) and testing (t_{tSet}) chunks. The heterogeneous learner uses different base-learning algorithms (b_{sLAlgo}) to extract meta-features ($M_{feat\,i}$). The meta model (M_{model}) uses $M_{feat\,i}$ for training and predicting a specific h_{lc}. The pretrained M_{model} is stored in $m_{odelRepo}$ and retrieved when required, which takes the device modality matrix (d_{mtx}) as an input and predicts the corresponding h_{lc}. The predicted value (such as sitting), along with d_{mtx} (accelerometer coordinates (x,y,z), gyroscope coordinates (x,y,z)), $l_{hRule}[c_{oncl}]$ (activity), and $l_{hRule}[c_{ond}]$ (accelerometer and gyroscope), is mapped in predicted matrices (p_{ls}). The p_{ls} depends on the input modalities, such as single (activity=sitting) or multiple (activity=sitting, location=living room) modalities. The decision maker analyzes p_{ls} and retrieves the relevant rules regarding the predicted context from $u_{prefBehr}$. Then, the weight of each rule is calculated based on the number of matched attributes, and the highest weighted rule is selected as a recommendation (r_{ec}), such as IF (activity=sitting, location=living) THEN (action=adjust temperature). Furthermore, the p_{ls} gets filter based on the selected r_{ec}. If multiple rules have the same weight, then all the rules are selected as a r_{ec} along with its corresponding p_{ls} that satisfy the context conditions (such as adjust temperature and play music).

E. PERSONALIZED REPOSITORY

The personalized repository validates the generated r_{ec} via user feedback (u_{srFB}) and maintains repositories such as modality data (P_{Repo}) and behavioral rules (b_{ehRepo}). Algorithm 5 describes our proposed approach for the personalized repository layer. The active learner interacts with the user and acquires feedback for each r_{ec} (r_{ec}: IF (activity=sitting, location=living) THEN (action=adjust temperature), u_{srFB}: IF(activity=sitting, location=dining) THEN (action=play music)). The u_{srFB} supports in conflicts (c_{onfl}) identification of the generated r_{ec} (c_{onfl}: location=dining, action=play music), which gets updated based on the identified c_{onfl}. Moreover, if c_{onfl} occurs in the condition of r_{ec} ($r_{ec}[c_{ond}]$: location=dining), then the corresponding predicted label is updated in the p_{ls} (such as replace 'dining' with 'living' for 'location' (h_{lc})). The updated p_{ls} and r_{ec} are stored in P_{Repo} and b_{ehRepo}, respectively.

Algorithm 5 Personalized Repository Layer

Input: r_{ec}, u_{srFB}, p_{ls}
Output: r_{ec}

foreach *r in* r_{ec} **do**
  $u_{srFB} \leftarrow$ acquire u_{srFB}
  if u_{srFB} *conflicts with* $r_{ec}[r]$ **then**
    $c_{onfl} \leftarrow$ identify c_{onfl} between u_{srFB} and $r_{ec}[r]$
    $r_{ec} \leftarrow$ update $r_{ec}[r]$ based on c_{onfl}
    if c_{onfl} *in* $r_{ec}[c_{ond}]$ **then**
      $p_{ls} \leftarrow$ update $p_{ls}[p]$ based on c_{onfl}
store p_{ls} in P_{Repo}
store r_{ec} in b_{ehRepo}
trigger r_{ec}

F. SERVICE PROVIDER

The service provider analyzes the input request and provides different healthcare services, such as telemedicine, education, e-coaching, smart pharmacies, smart hospitals, and mobile health services. Algorithm 6 describes our proposed approach for service providers. This layer analyzes the service request (S_{vReq}) based on the input modalities and then validates the consumer identity (c_{xID}) to ensure that only legitimate users can utilize the services and checks for ongoing session identifiers (S_{essID}). The S_{essID} is affiliated with the c_{xID} and S_{vReq}. If the session is expired/not initiated, then the S_{essID} is set as null, and the S_{vReq} is inferred after the generation of a new S_{essID}. For an ongoing S_{essID}, the S_{vReq} is inferred with the same S_{essID}. In the case of invalid c_{xID} and S_{essID}, the reauthentication request is generated to prevent unauthorized access and set the service response (S_{vRes}) as null.

Algorithm 6 Service Provider Layer

Input: S_{vReq}
Output: S_{vRes}
while S_{vReq} *is true* **do**
  $c_{xID} \leftarrow$ check c_{xID}
  $S_{essID} \leftarrow$ check S_{essID}
  if c_{xID} *is valid &* S_{essID} *is null* **then**
    initiate session
    $S_{essID} \leftarrow$ generate S_{essID}
    $S_{vRes} \leftarrow$ infer S_{vReq}
  else if c_{xID} *is valid &* S_{essID} *exists* **then**
    $S_{essID} \leftarrow$ use existing S_{essID}
    $S_{vRes} \leftarrow$ infer S_{vReq}
  else
    $S_{vRes} \leftarrow$ null
    re-authentication request

G. DATA MIGRATOR

The data migrator migrates the data from the source fog node (S_{fogN}) to the targeted cloud platform (T_{cloud}). S_{fogN} provides assistance to T_{cloud}, which contains limited resources (such as storage) and requires an efficient mechanism to improve the performance of health fog, as described in Algorithm 7. The data synchronizer depends on the migration time interval (m_{igrT}) and data-migration-flag (d_{migrF}). The m_{igrT} used for periodic data transfer is based on time (t). d_{migrF} is activated when limited storage is available in S_{fogN}. Upon activation, the trusted third party (TTP) authenticates the S_{fogN} and T_{cloud}. Then, the key generator (K_{gen}) is requested to generate and share the cryptographic keys among the communicating entities, which include symmetric keys (k) and asymmetric keys (such as public keys (p_k) and private keys (s_k)). Furthermore, the data synchronizer synchronizes the fog node repositories (f_{NRepo}) to ensure that the ongoing process is completed before data encryption. The f_{NRepo} includes

Algorithm 7 Data Migrator Layer

Input: d_{retrv}
Output: E_d
if $(t = m_{igrT})$ *or* (d_{migrF} *gets activated*) **then**
  $a_{uthFlag} \leftarrow$ *TTP* authenticate S_{fogN} & T_{cloud}
  if $a_{uthFlag}$ *is true* **then**
    K_{gen} share $s_{k\,S_{fogN}} || p_{k\,S_{fogN}} || p_{k\,T_{cloud}} || k$ with S_{fogN}
    K_{gen} share $s_{k\,T_{cloud}} || p_{k\,T_{cloud}} || p_{k\,S_{fogN}} || k$ with T_{cloud}
    $d_{retrv} \leftarrow S_{ync}$ data from f_{NRepo}
  foreach *data d in* d_{retrv} **do**
    $E_d \leftarrow E_{ncrypt}$(k,d)
    $D_{sig} \leftarrow E_{ncrypt}(s_{k\,S_{fogN}}, c_{ryptoHash}(E_d))$
    $E_{D_{sig}} \leftarrow E_{ncrypt}(p_{k\,T_{cloud}}, D_{sig})$
    $E_d \leftarrow E_d || E_{D_{sig}}$
    $d_{isprFlag} \leftarrow$ dispatch E_d to T_{cloud}
    if $d_{isprFlag}$ *is true* **then**
      remove d from f_{NRepo}

M_{apRepo}, $m_{odelRepo}$, a_{pRepo}, p_{pRepo}, d_{sRepo}, P_{Repo}, b_{ehRepo}, and $e_{logRepo}$. The data encrypter encrypts each repository with k to ensure confidentiality and efficiency. Then, the digital signature (D_{sig}) is computed and encrypted with $p_{k\,T_{cloud}}$ to ensure data integrity. The data dispatcher dispatches the encrypted data (E_d) to T_{cloud}, together with the encrypted digital signature ($E_{D_{sig}}$). Moreover, it analyzes the dispatcher flag ($d_{isprFlag}$) to remove the corresponding data from f_{NRepo}. The $d_{isprFlag}$ identifies the status of dispatched data (such as $d_{isprFlag}$ = true, which means that the data were successfully sent without tampering).

H. SECURITY MANAGER

This layer manages the security requirement of the health fog and ensures prevention against different attacks, such as masquerading, man-in-the-middle (MITM), eavesdropping, message tampering, and replay. The security manager analyzes the communication protocol (c_{ommP}) and ensures transport layer security. It also provides different services to the health fog, including authentication, access control, key management, confidentiality, and integrity. These services depend on the protocol and requirements of the health fog. Algorithm 8 describes the generic approach for security managers, which analyzes the security service request ($S_{ecSvReq}$) and forwards it to the relevant function for security service response ($S_{ecSvRes}$). To use our proposed health fog framework, the user needs to complete the registration (r_{eg}) and use the credentials for authentication (a_{uth}). We have defined a limit on authentication attempt (a_{uthAtt}) to avoid brute-force attacks. Access control (a_{ctrl}) identifies the rights of authenticated users and provides services accordingly. Key management (K_{mgt}) generates, shares, and revokes

Algorithm 8 Security Manager Layer

Input: $S_{ecSvReq}$
Output: $S_{ecSvRes}$
while $S_{ecSvReq}$ *is true* **do**
  if $S_{ecSvReq}$ *is* r_{eg} **then**
    $S_{ecSvRes} \leftarrow$ infer $S_{ecSvReq}$
  else if $S_{ecSvReq}$ *is* a_{uth} **then**
    if $a_{uthAtt} \leq$ *limit* l **then**
      $S_{ecSvRes} \leftarrow$ infer $S_{ecSvReq}$
      increment a_{uthAtt}
  else if $S_{ecSvReq}$ *is* a_{ctrl} **then**
    $S_{ecSvRes} \leftarrow$ infer $S_{ecSvReq}$
  else if $S_{ecSvReq}$ *is* K_{mgt} **then**
    $S_{ecSvRes} \leftarrow$ infer $S_{ecSvReq}$
  else if $S_{ecSvReq}$ *is* $c_{onfMeas}$ **then**
    $S_{ecSvRes} \leftarrow$ infer $S_{ecSvReq}$
  else if $S_{ecSvReq}$ *is* I_{meas} **then**
    $S_{ecSvRes} \leftarrow$ infer $S_{ecSvReq}$
  else
    $S_{ecSvRes} \leftarrow$ null

Algorithm 9 Event Monitor Layer

Input: e_{data}, c_{evt}
Output: e_{alert}
foreach d_{mtx} *in* $d_{mtxMap}[d_{mtx}]$ **do**
  $e_{data}[d_{mtx}] \leftarrow$ analyze e_{data} for time t
  $t_{hld}[d_{mtx}] \leftarrow$ set t_{hld} for $e_{data}[d_{mtx}]$
$\forall\, d_{mtx}$, **if** $c_{evt}[d_{mtx}] > t_{hld}[d_{mtx}]$ **then**
  $e_{alert}[d_{mtx}] \leftarrow$ generate e_{alert}
store e_{data}, t_{hld}, and e_{alert} in $e_{logRepo}$

cryptographic keys (such as k, s_k, and p_k). These cryptographic keys can be used for encryption/decryption and digital signatures to ensure confidentiality ($c_{onfMeas}$) and integrity (I_{meas}), respectively.

I. EVENT MONITOR

The event monitor analyzes the health fog operation for intrusion detection and generates an alert. Algorithm 9 describes the proposed approach for event monitoring, which analyzes the event data (e_{data}) for a specific time (t) and identifies the threshold (t_{hld}) for each d_{mtx}. The e_{data} consists of multiple logs, such as application, user activity, device, and compliance, which detect intrusion at different levels and support the design of health fog preventive strategies. The event manager compares the current event data (c_{evt}) with the identified t_{hld} to generate an event alert (e_{alert}) for necessary action. Then, the e_{data}, t_{hld}, and e_{alert} are stored along with the timestamp in the event log repository ($e_{logRepo}$), which can be used for audit and anomaly prediction in the future.

J. ADAPTIVE MODEL TUNING

The proposed health fog framework maintains the personalized repository that evolves the annotated dataset and finetunes the machine learning model after a specific time interval. We have considered the publicly available annotated data as a seed for predicting a specific context based on the input modalities. The personalized repository contains an active learner that interacts with the user and acquires feedback to refine the data labels. Algorithm 10 describes our proposed approach for adaptive model tuning, which periodically analyzes the t_{hld} of P_{Repo} for labeled data. If the stored data meet the t_{hld} condition, the corresponding a_{nnData} are retrieved from the d_{sRepo} and appended with the P_{Repo} in such a way that each feature contains its relevant values. Then, the updated a_{nnData} are stored in d_{sRepo} and the hybrid ensemble learner is requested to retrain the model. Similarly, $u_{prefBehr}$ is updated based on the b_{ehRepo} that supports the personalized recommendations.

Algorithm 10 Adaptive Model Tuning

Input: P_{Repo}, b_{ehRepo}
Output: update d_{sRepo} and a_{pRepo}
/* Analyze P_{Repo} and b_{ehRepo} */
foreach h_{lc} *in* $P_{Repo}[p_{ls}]$ **do**
  if *data* $d \geq t_{hld}$ **then**
    $a_{nnData} \leftarrow$ retrieve a_{nnData} from d_{sRepo},
    where $a_{nnData} \equiv P_{Repo}[p_{ls}]$
    $a_{nnData} \leftarrow$ map and append d with a_{nnData}
    store a_{nnData} in d_{sRepo}
    update model using a_{nnData}
  else if *rule r exists in* b_{ehRepo} **then**
    update r in $a_{pRepo}[u_{prefBehr}]$
  else
    periodic analysis continue

IV. EVALUATION AND RESULTS

In this section, we have evaluated our proposed secure health fog framework in terms of security, performance, and accuracy. To evaluate security protocols such as the Zigbee Secure Health Fog (ZigbeeSHF) and data migration, we used an automated formal verification tool Scyther [45] and AVISPA [46], which evaluate the protocol based on the threat model and exploit the vulnerabilities in different attacking scenarios. To evaluate the performance and accuracy, we deployed our proposed secure health fog framework in a 40 square meter smart studio apartment, which contained different sensors such as magnetic switches (detecting door openings), occupancy sensors (detecting indoor motion and controlling light switches), and temperature sensors (measuring heat energy). Each of these sensors was assigned a unique

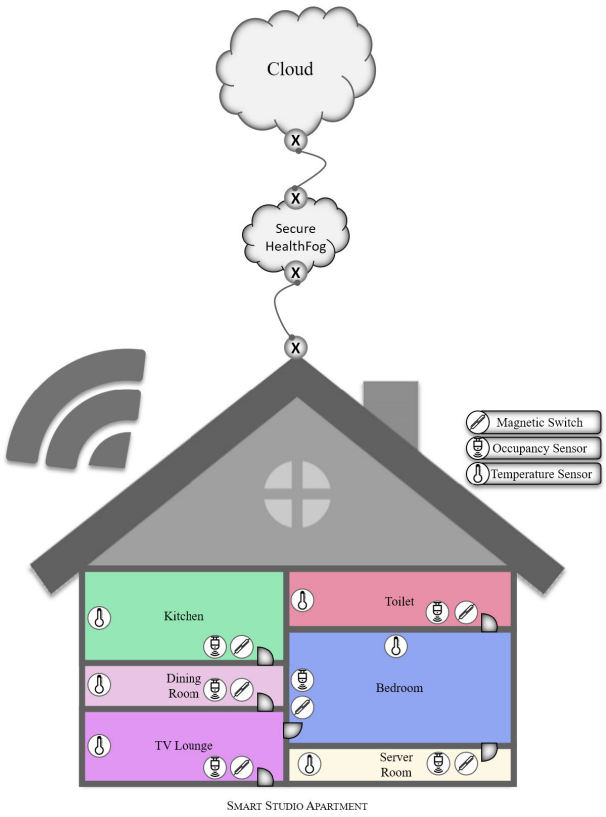

FIGURE 2. Smart studio apartment layout and sensors placement.

tag supporting data collection and event identification at a specific portion. Moreover, these sensors are customizable in the Zigbee family (CC1352R, CC1352P), which contains 352 kB of memory, 80 kB of RAM, and support security accelerators. Figure 2 presents the smart studio apartment layout and placement of sensors connected with the secure health fog and cloud. The description and result of each evaluation criterion is described as follows.

A. SECURITY EVALUATION

The secure health fog collects the data from IoT devices and maintains a personalized repository based on user feedback, which evolves the model and supports personalized context-sensitive decision making. The data collected from these devices contain personally identifiable information, and the exposure of such data to an unauthorized person may lead to severe consequences. For this purpose, we have proposed protocols to ensure wireless personal area network security and data migration security.

1) WIRELESS PERSONAL AREA NETWORK SECURITY

The smart studio apartment consists of magnetic switches, occupancy, and temperature sensors, which rely on the Zigbee standard and create a mesh networking topology, where multiple end devices connect with a single network coordinator [47]. Zigbee is popular due to its low cost, low energy consumption, and built-in security features that support a wide

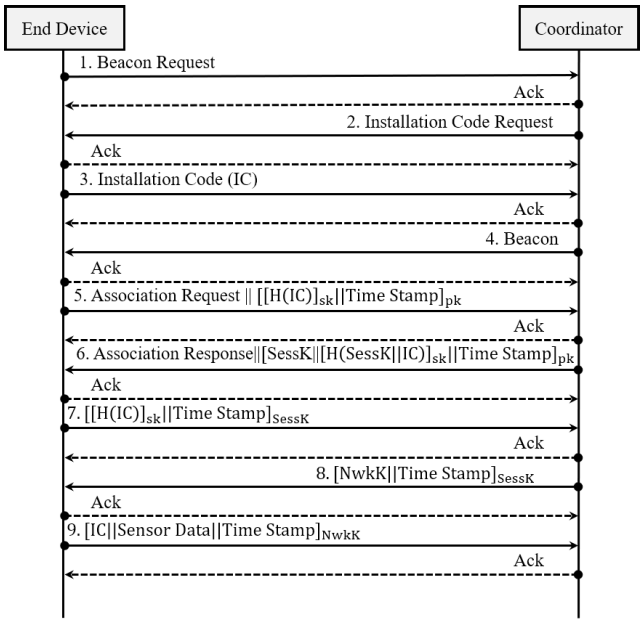

FIGURE 3. Zigbee Secure Health Fog (ZigbeeSHF) protocol for wireless personal area network security (sk: private key, pk: public key, SessK: session key, H: hash function, NwkK: Network Key).

range of home automation products and industrial devices. However, with all these benefits, Zigbee is still vulnerable to several attacks, such as device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay (as discussed in sections II-C and II-D). The sensors deployed in the smart studio apartment rely on Zigbee 3.0, which derives a unique link key from the installation code using AES-MMO and uses it to encrypt the network key instead of the global master key. However, this approach has a vulnerability; a unique link key exists and is transmitted in plain text. If the adversary obtains this link key, then the security of the Zigbee network is compromised. Therefore, we have proposed the Zigbee Secure Health Fog (ZigbeeSHF) protocol to improve the installation code mechanism by considering it a sensor identifier and ensure prevention against the mentioned attacks. Figure 3 illustrates the interaction between the Zigbee end device and coordinator. Additionally, the explanation of each step is described as follows.

1) Initially, the end device broadcasts a beacon request until it receives a response from the Zigbee coordinator.
2) When the Zigbee coordinator in the open state receives a beacon request, it generates an installation code request to obtain the sensor identifier.
3) The installation code affiliates with a specific end device and is sent to the coordinator through physical or remote access.
4) Upon receiving an installation code, the coordinator broadcasts a beacon that includes a media access control (MAC) address and a personal area network (PAN) identifier.

FIGURE 4. ZigbeeSHF protocol verification result using Scyther.

5) Based on the provided information, the end device generates an association request, which is concatenated with the installation code digital signature and timestamp and encrypted with the end device public key.
6) The coordinator verifies and validates the received digital signature. Upon success, an association response and a session key are generated. To ensure integrity, the digital signature of the session key and installation code are computed. Then, the digital signature, session key, and timestamp are encrypted with the end device public key.
7) The end device receives the encrypted packet from the coordinator and validates the session key. To prove the identity again, the end device computes the hash value of the installation code, appends with the timestamp, and then encrypts it with the received session key.
8) The coordinator verifies the end device's identity and then transmits the network key and timestamp encrypted with the shared session key.
9) When the end device wants to transmit the data over the Zigbee network, it uses the network key to encrypt the data along with a timestamp.

Our proposed ZigbeeSHF considers identity and data authenticity, integrity, data consistency, and confidentiality, ensuring prevention against device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay. To support our claims, we used the automated formal verification tool Scyther. Figure 4 presents the ZigbeeSHF protocol verification results evaluated using Scyther, and the description of these claims are described as follows.

- Secret: ZigbeeSHF verifies secret claims because the session key, network key, and sensor data are transmitted securely over the communication channel, which prevents eavesdropping and privacy leakage.
- Alive: This claim ensures the integrity of transmitted data over the communication channel, which indicates that ZigbeeSHF prevents message tampering.
- Weakagree: Most attacks are possible due to a lack of authentication. The weakagree claim ensures that ZigbeeSHF provides source identity authentication and source data authentication because it uses digital signatures during the association request and association response. This ensures prevention against different attacks, such as those of man-in-the-middle, masquerading, and malicious insider.
- Commit: ZigbeeSHF ensures correct response on running events, specifically the exchange of session and network keys among the communicating entities, supporting device control and preventing data loss.
- Niagree: The ZigbeeSHF communicating entities agreed upon the data values and verified the noninjective agreement (Niagree), which ensures prevention against fake device injection.
- Nisynch: ZigbeeSHF ensures noninjective synchronization (Nisynch) and considers the timestamp as a nonce, which prevents replay attacks because it is transmitted securely among communicating entities.

Moreover, we have also evaluated ZigbeeSHF with AVISPA, which analyzes the protocol based on secrecy, authentication, proof-of-origin, and integrity [46], [48]. The AVISPA works on formal methods principles to achieve the security goals, which includes On-the-Fly Model-Checker (OFMC) [49], Constraint Logic-based Attack Searcher (CL-ATSE) [50], Satisfiability-based Model Checker (SATMC) [51], and Tree Automata-based Protocol Analyzer (TA4SP) [52]. We evaluated ZigbeeSHF with OFMC, which identifies all known attacks from the Clark and Jacob library. Figure 5 presents the output of AVISPA, which summarized ZigbeeSHF as SAFE based on OFMC analysis and ensured prevention against the Dolev-Yao intruder.

2) DATA MIGRATION SECURITY

The secure health fog contains limited resources and required data migration to the cloud for long-term storage (as described in section III-G). The procedure starts when the data-migration flag (d_{migrF}) is activated or the migration time interval (m_{igrT}) ends. Then, the fog node requests the cloud for mutual authentication based on access tokens and security credentials. After successful authentication, the fog node encrypts the data and timestamp using symmetric key cryptography. Additionally, the digital signature of the encrypted data is computed along with the timestamp and then encrypted with the cloud public key. Finally, concatenate both blocks and transmits them over the communication

TABLE 3. Entities, function, key size, and key generation time of Secure Health Fog framework.

Entities	Function	Key Size (bit)	Key Generation Time (ms)
Symmetric Key Algorithm	Advanced Encryption Standard (AES)	256	0.0015
Public Key Algorithm	Elliptic Curve Cryptography (ECC)	160	513.096
Hash Function	Secure Hash Algorithms (SHA)	**Digest Length:** 256 bit	

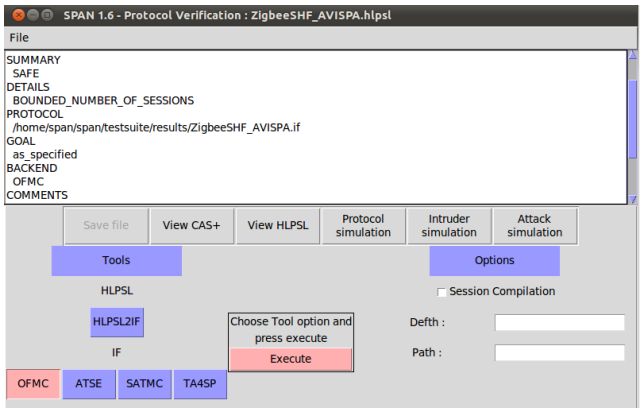

FIGURE 5. ZigbeeSHF protocol verification result using AVISPA.

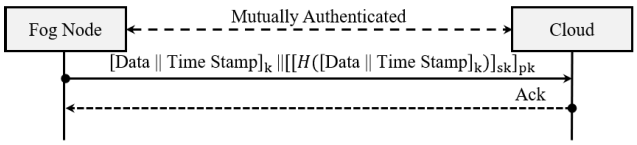

FIGURE 6. Proposed data migration protocol (k: symmetric key,sk: private key, pk: public key, H: hash function).

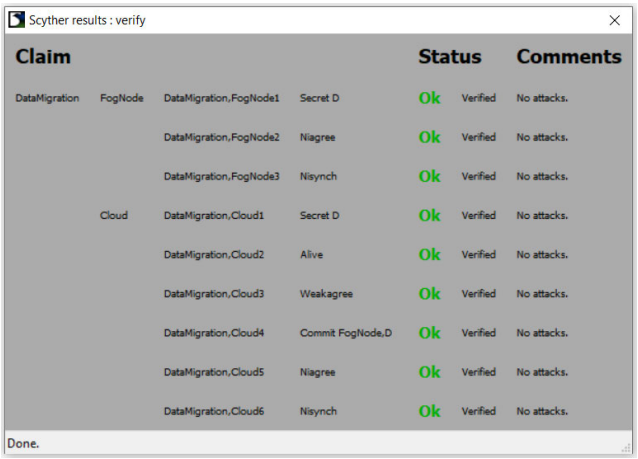

FIGURE 7. Data migration protocol verification result using Scyther.

channel, as shown in Figure 6. The purpose of encrypting the data using symmetric key cryptography is to ensure data confidentiality and reduce computational overhead. The secure digital signature of encrypted data ensures data consistency and authenticity, and identifies tampering during data migration. Similar to ZigbeeSHF, we formally verified the protocol using Scyther to identify vulnerabilities. Figure 7 presents the verification results obtained from Scyther. We have encrypted

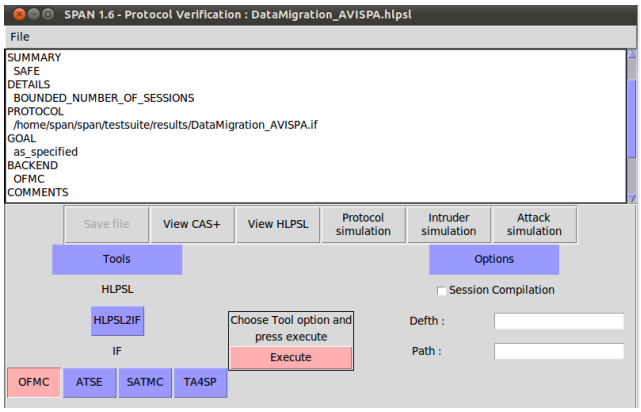

FIGURE 8. Data Migration protocol verification result using AVISPA.

the data using symmetric key cryptography that ensures data confidentiality and verifies the secret claim. The verification of Niagree and Nisynch ensures prevention against man-in-the-middle and replay attacks, respectively. Similarly, alive, weakagree, and commit verification ensures data integrity, data consistency, source identity authentication, and source data authentication due to secure digital signatures. Figure 8 describes the protocol verification result using AVISPA, which analyzes the secrecy, authentication, proof-of-origin, and integrity of the data migration protocol under the OFMC and summarizes it as SAFE.

B. PERFORMANCE MEASURE

The secure health fog considers various factors to provide reliable personalized services. However, efficiency is considered an essential factor for measuring the performance of information security systems. Therefore, we have evaluated our proposed scheme's performance in terms of time efficiency and processing overhead. Table 3 summarizes the key generation time of cryptographic functions used in the secure health fog framework. We have used the advanced encryption standard (AES) with a 256 bit key size to encrypt a large amount of data and reduce the computational overhead. The AES-256 bits take an average of 0.0015 milliseconds for a one-time key generation. Moreover, we used elliptic curve cryptography (ECC) with a 160 bit key size for a relatively small amount of data encryption and digital signature, specifically for the secure transmission of a secret key. The ECC-160 bits take an average of 513.096 milliseconds for key pair generation and distribution of the public key. We used the secure hash algorithm (SHA) to generate a collision-resistant cryptographic hash value of 256 bits to ensure data integrity.

To evaluate the computational time required during the Zigbee joining procedure, we have identified that the sensors deployed in our smart studio apartment rely on Zigbee 3.0, which supports the installation code feature. Therefore, we analyzed the computational time of Zigbee 3.0 with and without installation code. Then, we deployed our proposed ZigbeeSHF in the smart studio apartment using Zigbee hardware belonging to CC1352R and CC1352P. According to our analysis, Zigbee 3.0 with and without installation code takes 0.3297 seconds and 0.2673 seconds, respectively, while our proposed ZigbeeSHF takes 0.3581 seconds. Figure 9 presents the difference between the computational time and the time taken by each protocol. ZigbeeSHF exchanges three types of encrypted digital signatures during the association phase, which requires computational time to verify the end device identity and receive requests. Therefore, it required 7.93% and 25.35% more computation time than Zigbee with and without installation code, respectively.

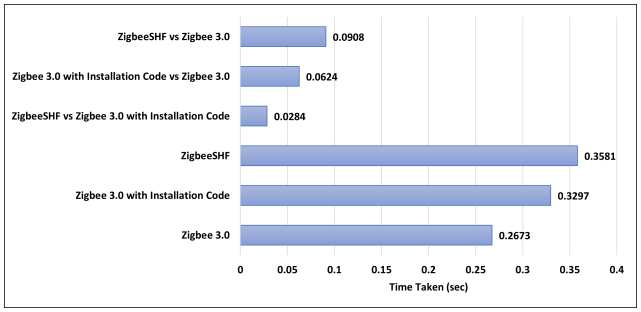

FIGURE 9. Comparison of ZigbeeSHF with the standard Zigbee 3.0 Protocol.

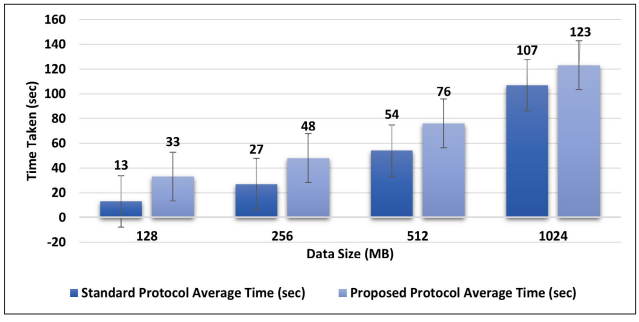

FIGURE 10. Data migration time taken with transfer rate between 80 Mbps and 100 Mbps.

Similarly, we evaluated our proposed data migration protocol based on different data sizes, as shown in Figure 10. The result shows that the proposed data migration protocol takes 36.57% more computational time than the standard protocol due to the concatenation of the encrypted digital signature, with a transfer rate between 80 Mbps and 100 Mbps. Moreover, Figure 11 describes the CPU utilization for 1 minute during the peak interval, where the secure health fog collects the data from end devices, trains a personalized model, and migrates the data to the cloud simultaneously. The result shows that the average CPU utilization of the secure health

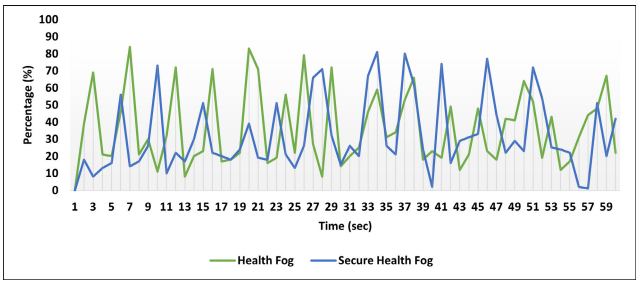

FIGURE 11. Health Fog and Secure Health Fog CPU utilization for 1 minute.

fog is 3% less than that of the standard health fog because the secure health fog utilizes the resources efficiently and reliably.

C. ACCURACY EVALUATION

Accuracy identifies the effectiveness of an algorithm based on the probability of true values. We have set up an environment to evaluate the accuracy of our proposed secure health fog framework and identify the impact of adaptive model tuning. The details of the accuracy evaluation are described as follows.

1) DATASET SELECTION, PREPROCESSING AND ENSEMBLE LEARNING

To evaluate the accuracy of our proposed secure health fog framework, we considered two datasets: publicly available data and smart studio apartment data. The purpose of using the publicly available dataset was to identify the effectiveness of adaptive model tuning on personalized recommendations. We selected the data from the CASAS project based on features that are similar to those of the smart studio apartment data. The publicly available data from the CASAS project consist of data collected from 30 smart homes affiliated with different age group residents performing daily routine activities [53]. Our smart studio apartment dataset consists of one-month data collected from 25-35 age group residents performing daily routine activities. We have used similar labels as the CASAS project and performed several preprocessing steps to ensure consistency among both datasets [54]. Moreover, we mapped and synchronized each dataset in a uniform time grid, in which one instance represents a day with varying length and activities. Additionally, we analyzed the datasets to extract the relationship of low-level and high-level contexts, which was stored in the mapping repository (M_{apRepo}). According to [55], human behavior evolves around three states: physical activity, sedentary, and sleep. Therefore, we mapped the activity labels to a higher level based on granularity into these three categories without overlapping, as shown in Figure 12. For ensemble learning, we used five base learners, support vector machines (SVMs), artificial neural networks (ANNs), bagging prediction, random forest, and least squares boosting (LSBoost), with the evolutionary salp swarm algorithm (SSA). The approach is

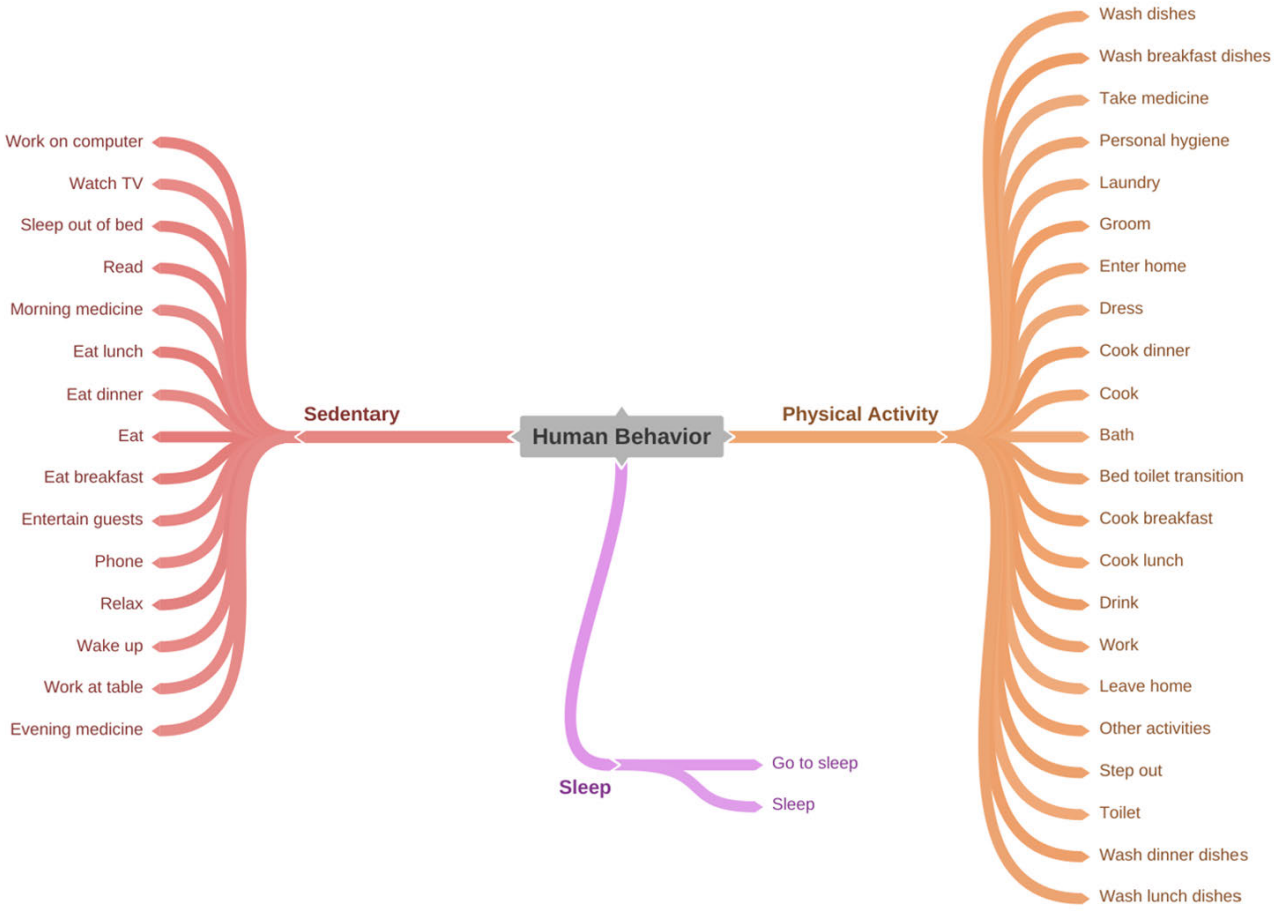

FIGURE 12. Categorization of datasets into physical activity, sedentary, and sleep based on human behavior state.

similar to the concept proposed in [56], where SSA optimizes the weight and solves the electromagnetic problem. We have applied this approach to predict the user activities inside the smart studio apartment based on CASAS project data and smart studio apartment data.

2) INTERACTIVE MEDIUM SELECTION AND PARTICIPANT GUIDANCE

Our proposed secure health fog framework collects user preferences and feedback to maintain a personalized repository for adaptive model tuning. Therefore, the interface needs to be user-friendly and appropriate for any age group. For this purpose, we have used an extended version of our previous work, Medical Instructed Real-time Assistant (MIRA), which interacts with the user in a natural way of communication and supports speech/text for interactive conversation [57]. However, instead of identifying the patient's health condition, we used MIRA for user preferences and feedback collection and named it MIRA extension ($MIRA_{ext}$). Moreover, we selected the participant after a standard operating procedure provided by the Kyung Hee University Ethics Assessment Committee (KHU-EAC). The participant belonged to the 35-45 age group and had underlying chronic conditions such as diabetes and hypertension. Written consent was signed by the participant, which were informed that the collected data would be used for research purposes only, and no personally identifiable information would be released under any circumstances. Additionally, the collected information will be destroyed after five years based on the KHU-EAC policy. Furthermore, we guided the participant to perform their daily routine activities inside the smart studio apartment and acquire a feedback schedule, which contains dynamic time intervals, and configured the application accordingly to generate an alert. Initially, $MIRA_{ext}$ acquired user preferences in an IF-THEN format, consisting of two categories: user behavior and privacy preservation. User behavior describes the nature of an individual that needed to develop policies for generating a recommendation. Privacy preservation ensures data ownership to specify which type of data needs to be stored on the cloud and what type of data needs to be discarded. Based on the user preferences, the secure health fog framework makes a decision about the actions.

3) IMPACT OF ADAPTIVE MODEL TUNING ON ACCURACY

We analyzed the participant's daily activity inside the smart studio apartment for 30 days and maintained a catalog, which

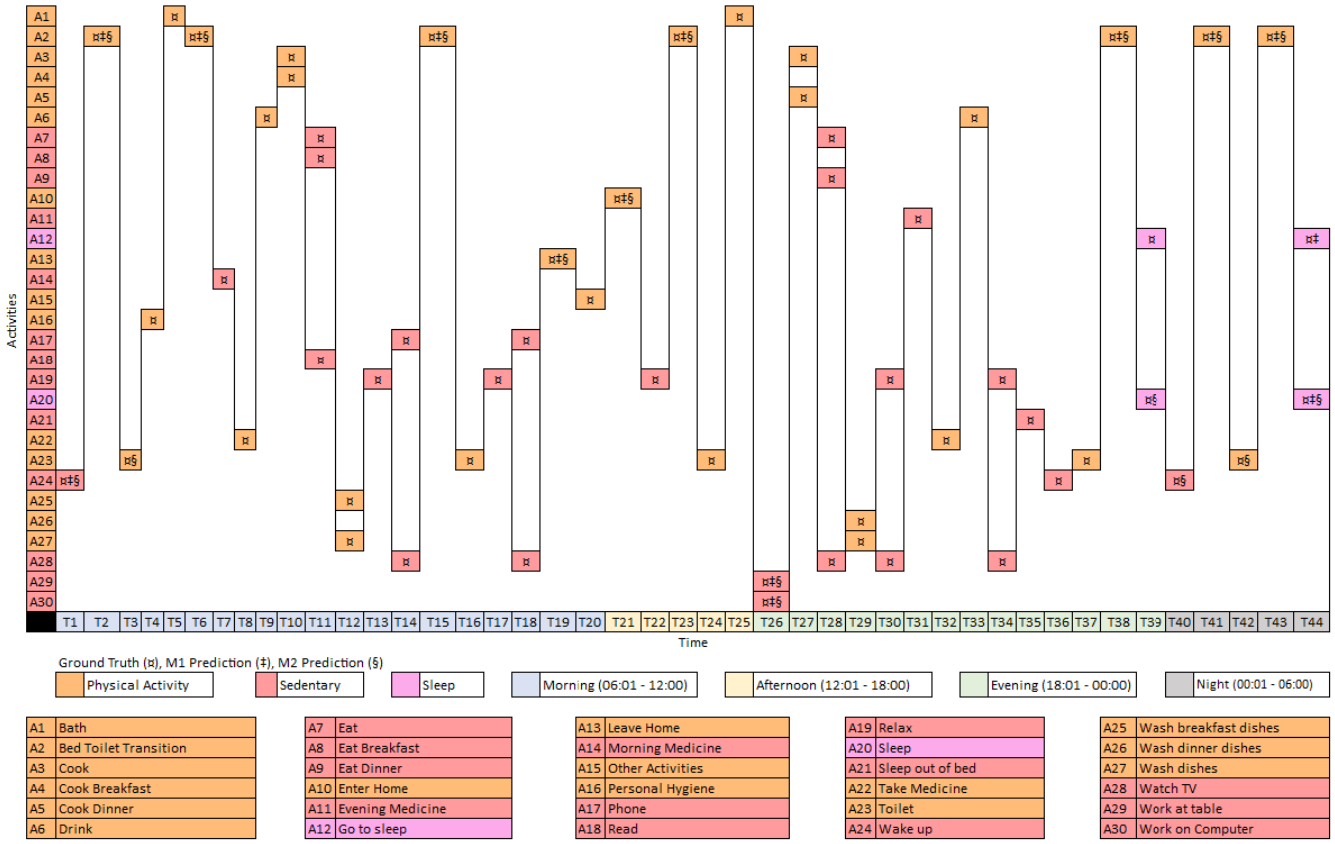

FIGURE 13. Day 1 transition state of participant activities inside smart studio apartment.

included ground truth, predicted activity, activity state, and timestamp data. Figure 13 describes the transition state of the participant for the first 24 hours inside the smart studio apartment. The x-axis represents the time, while the y-axis describes the user activities on a specific day. The different colors identify the nature of activities, such as light orange for physical activity, rose for sedentary, and lavender for sleep. Similarly, we categorized the day into morning (blue: 06:01-12:00), afternoon (light yellow: 12:01-18:00), evening (green: 18:01-00:00), and night (light gray: 00:01-06:00). The ground truth (¤) data described the actual activity of the participant inside the smart studio apartment. M1 (‡) and M2 (§) represent the predicted activities based on pretrained models such as CASAS project data (M1) and smart studio apartment data (M2). Moreover, $MIRA_{ext}$ interacts with the participant as per the schedule, acquires feedback for each predicted activity, annotates the collected data and stores in the personalized repository for adaptive model tuning. For the proof of concept, we merged the personalized label data collected within 24 hours with the preprocessed data and retrained the models, which supported personalized decision making and evolved the models periodically after 24 hours. Figure 14 presents the day-wise accuracy evaluated based on user feedback. The x-axis describes the number of days and status of personalized labeled data (P). The y-axis presents the accuracy in percentage. According to our analysis, adaptive model tuning improves the accuracy of M1 (pretrained model of CASAS project data) and M2 (pretrained model of smart studio apartment data) by 36% and 29%, respectively. Furthermore, Figure 15 presents the category-wise accuracy of predicted activities based on M1 and M2. The x-axis describes the model with personalized label data, and the y-axis presents accuracy as a percentage for each category. The results show that M1 accurately predicted 46.81% physical activity, 21.84% sedentary, and 31.35% sleep. Similarly, M2 predicted 46.66% physical activity, 21.73% sedentary, and 31.61% sleep, which indicated that both models were correlated in terms of category-wise accuracy.

V. DISCUSSION

The proposed SHF framework was evaluated using security, performance and accuracy measurements. We considered the security of the Wireless Personal Area Network (WPAN) and data migration. The WPAN supports several protocols based on IoT device compatibility, including infrared data association (IrDA), Bluetooth, Z-Wave, and Zigbee. To identify the protocol of IoT devices in our smart studio apartment, we examined the wireless microcontroller model to define the scope of communication protocols, and then analyzed it using Wireshark to validate the protocol based on the header

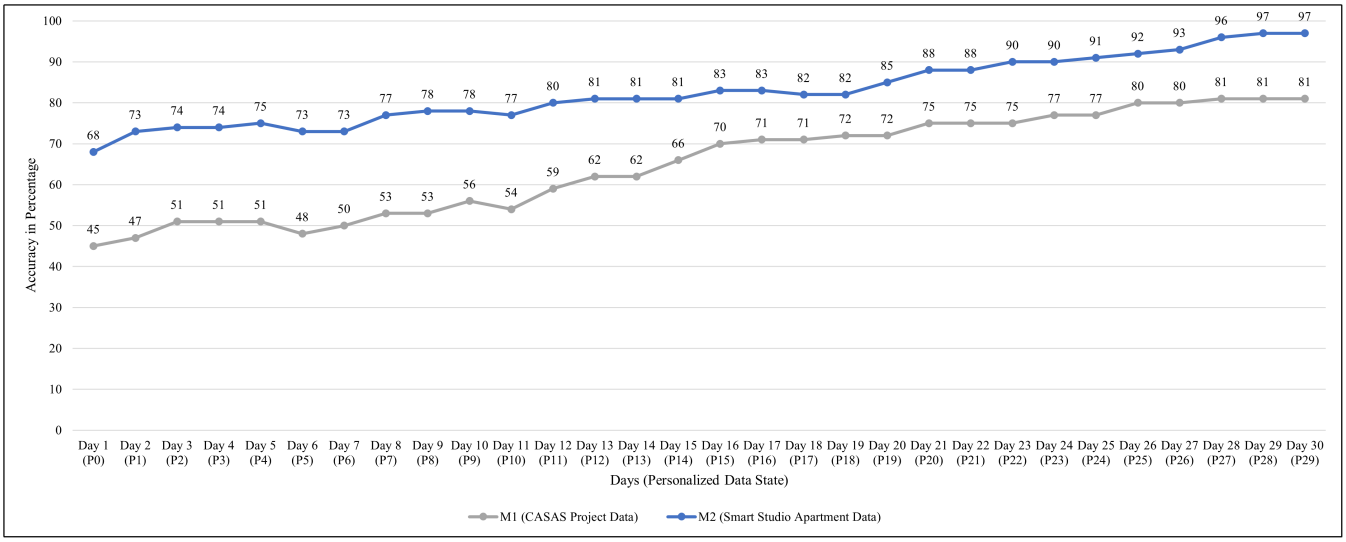

FIGURE 14. Day-wise accuracy based on user feedback.

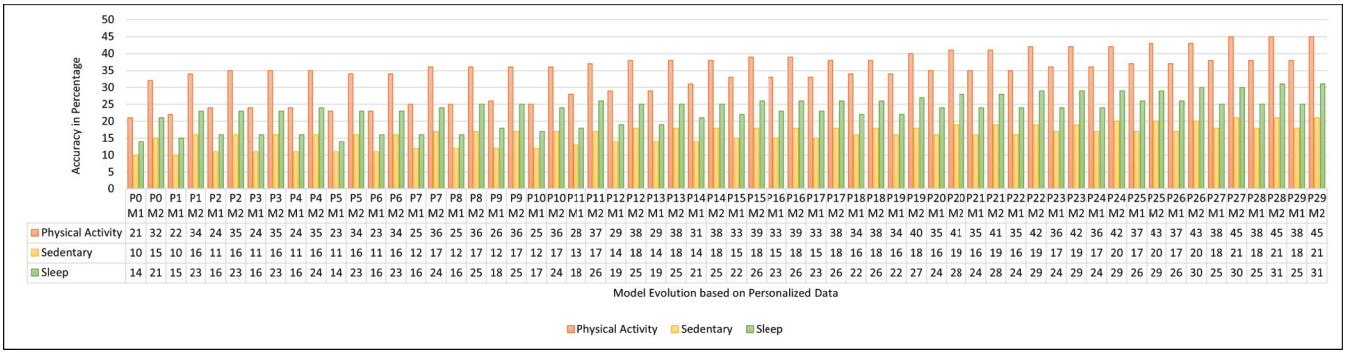

FIGURE 15. M1 and M2 day-wise accuracy based on human behavior state.

and payload. We found that the smart studio apartment IoT devices use the Zigbee 3.0 protocol on the CC13 × 2 microcontroller. Zigbee 3.0 uses installation code for authentication, which makes the protocol vulnerable to various attacks, such as device control, eavesdropping, fake device injection, malicious insiders, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay. Our proposed ZigbeeSHF uses ECC with a key size of 160 bits for source identity and data authentication, SHA-256 for data integrity, and AES-256 for data protection. The ZigbeeSHF considers the installation code as a device identifier and securely transmits the network key after two-factor authentication (shown in steps 5 and 7 of Figure 3). If the network key is compromised, the security of WPAN communication is at risk. The proposed ZigbeeSHF was evaluated using the formal verification tools Scyther and AVISPA, which categorize it as secure against the mentioned attacks. Similarly, the data migration protocol was enhanced with ECC, SHA-256 and AES-256 to ensure the authenticity, integrity and confidentiality of the data. Scyther and AVISPA summarize the data migration protocol as secure within the scope.

IoT devices consist of limited resources in terms of memory and computational power, which makes it challenging to propose an efficient and reliable security protocol. Therefore, we use the agile model to design and develop our proposed protocols, such as ZigbeeSHF and data migration. Initially, we have considered the Rivest-Shamir-Adleman (RSA) algorithm as public key cryptography with the key size of 512 bits, 1024 bits and 2048 bits which take 438.11 milliseconds, 686.08 milliseconds and 1317.63 milliseconds on average to generate and distribute the key pair. Similarly, we analyze the ECC with 160 bits, 224 bits and 256 bits keys which take 513.096 milliseconds, 851.56 milliseconds and 1184.129 milliseconds on average respectively. According to [58], [59] ECC consumes less battery resources and computational power, which can be considered as an efficient public key cryptography compared to RSA. Therefore, we have used ECC (160 bits) instead of RSA. In addition, the AES-256 is used to encrypt a large amount of data which is resistant to brute force attacks and reduces the computational overhead. The SHA-256 is used to quickly compute a collision-resistant hash value to ensure data integrity.

Based on these constraints, we found that the ZigbeeSHF end devices take 0.3581 seconds to join a Zigbee network and transmit the data. ZigbeeSHF exchanges three types of encrypted digital signatures, which require 7.93% and 25.35% more computation time than Zigbee with and without installation code, respectively. The data migration protocol concatenates the encrypted data along with the encrypted digital signature, which requires 36.57% more computation time than the standard protocol. The proposed protocols are comparatively expensive compared to the existing approaches, but this did not affect the overall performance of our proposed SHF framework. Moreover, the overall CPU utilization is 3% less than the standard health fog framework due to the appropriate selection and strategic placement of cryptographic algorithms.

We collected the participant feedback on predicted activities generated from pre-trained models of CASAS project data and smart studio apartment data. The participant was overwhelmed with continuous feedback and received an average of 97 and 51 notifications per day in the first (day 1∼15) and second (day 15∼30) half of the month, respectively. However, the collected feedback improves the annotated data labels, which supports adaptive model tuning and personalized recommendations. Each model evolves periodically with the personalized label data collected during 24 hours along with the preprocessed data. The result (Figure 14) shows that the accuracy of M1 (CASAS project data) and M2 (smart studio apartment data) without adaptive model tuning on day 1 was 45% and 68%, respectively, which gradually increases to 81% and 97% with the evolution of models after every 24 hours. On days 6 and 11, the participant performs some additional activities due to which the predicted labels were incorrect, and the accuracy decreases compared to the previous day. The accuracy of both models was unchanged after day 28, which shows the maximum accuracy achieved within 30 days for the selected datasets based on adaptive model tuning. Furthermore, the category-wise activity prediction (Figure 15) describes that both models predicted physical activity and sleep with high accuracy compared to sedentary. However, predicting sedentary activity with limited IoT devices such as magnetic switches, occupancy sensors, and temperature sensors is challenging. In the future, we will increase the scope of IoT devices and evaluate the accuracy of adaptive model tuning with relevant datasets.

VI. CONCLUSION

The health fog delivers healthcare as a fog service, which provides efficient services to the end user and improves their quality of life. According to our analysis, most of the existing studies have focused on performance measures and considered security as a secondary feature due to computational overhead, leading to several cyber-attacks. In this paper, we proposed a secure health framework that considered security, performance, and accuracy as a primary factor to ensure prevention against different attacks. Our proposed secure health fog maintains a personalized repository based on user feedback that evolves the model and supports personalized context-sensitive decision-making. Additionally, we proposed protocols for providing wireless personal area network security and data migration security. The proposed protocols were evaluated with Scyther and AVISPA, which ensures prevention against device control, eavesdropping, fake device injection, malicious insider, man-in-the-middle, masquerading, message tampering, privacy leakage, and replay. For proof of concept, we deployed our secure health fog framework in a smart studio apartment to evaluate the performance and accuracy. The results show that the proposed protocols were expensive compared to existing approaches, but it did not affect the overall performance. Moreover, adaptive model tuning was very effective and gradually improved the accuracy within a short period. Based on the user experience, continuous feedback and interaction were burdensome, but the feedback frequency decreased with the evolution of models.

The secure health fog framework was deployed in a smart studio apartment, which consists of magnetic switches, occupancy sensors, and temperature sensors. Therefore, we considered a dataset that consists of relevant features and recognizes the activities inside smart studio apartment. In the future, we will evaluate the proposed framework with the Internet of Medical Things (IoMT) and identify the effectiveness of adaptive model tuning. Additionally, we will design attack models to evaluate the security of secure health fog in a real environment. To evaluate the performance measure, we will extend the evaluation matrix in terms of processing time, energy utilization, power consumption, and latency.

REFERENCES

[1] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1321–1330, Mar. 2015.

[2] R. Akhavian and A. H. Behzadan, "Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers," *Adv. Eng. Informat.*, vol. 29, no. 4, pp. 867–877, Oct. 2015.

[3] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[4] O. DeMasi, S. Feygin, A. Dembo, A. Aguilera, and B. Recht, "Well-being tracking via smartphone-measured activity and sleep: Cohort study," *JMIR mHealth uHealth*, vol. 5, no. 10, p. e137, Oct. 2017.

[5] S. M. Alarcao and M. J. Fonseca, "Emotions recognition using EEG signals: A survey," *IEEE Trans. Affect. Comput.*, vol. 10, no. 3, pp. 374–393, Jul. 2019.

[6] M. Chen, Y. Ma, J. Song, C. F. Lai, and B. Hu, "Smart clothing: Connecting human with clouds and big data for sustainable health monitoring," *Mobile Netw. Appl.*, vol. 21, no. 5, pp. 825–845, 2016.

[7] J. Almahmoud, "Melodic trainer: An interactive musical approach for fitness training," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep., vol. 1001, p. 48104.

[8] Y. S. Delahoz and M. A. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, vol. 14, no. 10, pp. 19806–19842, 2014.

[9] B. B. Gibbs, A. L. Hergenroeder, P. T. Katzmarzyk, I.-M. Lee, and J. M. Jakicic, "Definition, measurement, and health risks associated with sedentary behavior," *Med. Sci. Sports Exerc.*, vol. 47, no. 6, p. 1295, 2015.

[10] E. Chiauzzi, C. Rodarte, and P. DasMahapatra, "Patient-centered activity monitoring in the self-management of chronic health conditions," *BMC Med.*, vol. 13, no. 1, p. 77, 2015.

[11] *World Population Prospects: The 2017 Revision, Key Findings and Advance Tables*, United Nations, Dept. Econ. Social Affairs, Population Division, New York, NY, USA, 2017.

[12] H. Zhan, L. Wang, S. Chen, P. M. Kumar, and P. M. Shakeel, "Detection and alerting system of nearby medical facilities during emergency using IoT sensors," *J. Ambient Intell. Hum. Comput.*, pp. 1–13, Mar. 2021.
[13] A. R. Hameed, S. U. Islam, I. Ahmad, and K. Munir, "Energy- and performance-aware load-balancing in vehicular fog computing," *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100454.
[14] W. Bai, Z. Ma, Y. Han, M. Wu, Z. Zhao, M. Li, and C. Wang, "Joint optimization of computation offloading, data compression, energy harvesting, and application scenarios in fog computing," *IEEE Access*, vol. 9, pp. 45462–45473, 2021.
[15] V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, "Smart items, fog and cloud computing as enablers of servitization in healthcare," *Sensors Transducers*, vol. 185, no. 2, p. 121, 2015.
[16] O. Fratu, C. Pena, R. Craciunescu, and S. Halunga, "Fog computing system for monitoring mild dementia and COPD patients—Romanian case study," in *Proc. 12th Int. Conf. Telecommun. Modern Satell., Cable Broadcast. Services (TELSIKS)*, Oct. 2015, pp. 123–128.
[17] X. Masip-Bruin, E. Marín-Tordera, A. Alonso, and J. Garcia, "Fog-to-cloud computing (F2C): The key technology enabler for dependable e-health services deployment," in *Proc. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2016, pp. 1–5.
[18] A. Monteiro, H. Dubey, L. Mahler, Q. Yang, and K. Mankodiya, "Fit: A fog computing device for speech tele-treatments," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 1–3.
[19] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog data: Enhancing telehealth big data through fog computing," in *Proc. ASE BigData SocialInformat. 2015*, 2015, p. 14.
[20] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Auton. Secure Comput., Pervasive Intell. Comput.*, Oct. 2015, pp. 356–363.
[21] J. K. Zao, T. T. Gan, C. K. You, S. J. R. Méndez, C. E. Chung, Y. T. Wang, T. Mullen, and T. P. Jung, "Augmented brain computer interaction based on fog computing and linked data," in *Proc. Int. Conf. Intell. Environ.*, Jun. 2014, pp. 374–377.
[22] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.
[23] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Generat. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.
[24] C. Thota, R. Sundarasekar, G. Manogaran, R. Varatharajan, and M. Priyan, "Centralized fog computing security platform for IoT and cloud in healthcare system," in *Fog Computing: Breakthroughs in Research and Practice*. Hershey, PA, USA: IGI Global, 2018, pp. 365–378.
[25] H. B. Hassen, N. Ayari, and B. Hamdi, "A home hospitalization system based on the Internet of Things, fog computing and cloud computing," *Informat. Med. Unlocked*, vol. 20, Jan. 2020, Art. no. 100368.
[26] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, "HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments," *Future Gener. Comput. Syst.*, vol. 104, pp. 187–200, Mar. 2020.
[27] P. H. Vilela, J. J. P. C. Rodrigues, P. Solic, K. Saleem, and V. Furtado, "Performance evaluation of a fog-assisted IoT solution for e-health applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 379–386, Aug. 2019.
[28] Y. Cao, S. Chen, P. Hou, and D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Aug. 2015, pp. 2–11.
[29] R. Craciunescu, A. Mihovska, M. Mihaylov, S. Kyriazakos, R. Prasad, and S. Halunga, "Implementation of fog computing for reliable e-health applications," in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2015, pp. 459–463.
[30] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 108–119, Jan./Mar. 2017.
[31] J. Vora, S. Kaneriya, S. Tanwar, S. Tyagi, N. Kumar, and M. S. Obaidat, "TILAA: Tactile internet-based ambient assistant living in fog environment," *Future Gener. Comput. Syst.*, vol. 98, pp. 635–649, Sep. 2019.
[32] T. N. Gia, I. B. Dhaou, M. Ali, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease," *Future Gener. Comput. Syst.*, vol. 93, pp. 198–211, Apr. 2019.
[33] S. K. Sood, V. Sood, I. Mahajan, and Sahil, "An intelligent healthcare system for predicting and preventing dengue virus infection," *Computing*, pp. 1–39, Jan. 2021.
[34] C. B. Liang, M. Tabassum, S. B. A. Kashem, Z. Zama, P. Suresh, and U. Saravanakumar, "Smart home security system based on ZigBee," in *Advances in Smart System Technologies*. Singapore: Springer, 2021, pp. 827–836.
[35] S. C. Ergen, "ZigBee/IEEE 802.15.4 summary," Dept. Elect. Eng. Comput. Sci., UC Berkeley, Berkeley, CA, USA, Tech. Rep., Sep. 2004, p. 11, vol. 10, no. 17.
[36] S. Khanji, F. Iqbal, and P. Hung, "ZigBee security vulnerabilities: Exploration and evaluating," in *Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS)*, Jun. 2019, pp. 52–57.
[37] L. Li, P. Podder, and E. Hoque, "A formal security analysis of ZigBee (1.0 and 3.0)," in *Proc. 7th Symp. Hot Topics Sci. Secur.*, Sep. 2020, pp. 1–11.
[38] *AN1089: Using Installation Codes With ZigBee Devices*. Accessed: Feb. 27, 2021. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1089-using-installation-codes-with-zigbee-devices.pdf
[39] W. Wang, F. Cicala, S. R. Hussain, E. Bertino, and N. Li, "Analyzing the attack landscape of ZigBee-enabled IoT systems and reinstating users' privacy," in *Proc. 13th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2020, pp. 133–143.
[40] R. Haakegaard and J. Lang. (2015). *The Elliptic Curve Diffie-Hellman (ECDH)*. [Online]. Available: https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf
[41] S. Okada, D. Miyamoto, Y. Sekiya, and H. Nakamura, "Proposal for LDOS attack using indirect transmission in ZigBee and a countermeasure against it," *IEICE Tech. Rep.*, vol. 120, no. 413, pp. 179–184, 2021.
[42] P. Tedeschi, S. Sciancalepore, A. Eliyan, and R. D. Pietro, "LiKe: Lightweight certificateless key agreement for secure IoT communications," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 621–638, Jan. 2020.
[43] *CC1352R Simplelink High-Performance Multi-Band Wireless MCU*. Accessed: Apr. 13, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc1352r.pdf?ts=1618227824617
[44] *CC1352P Simplelink High-Performance Multi-Band Wireless MCU With Integrated Power Amplifier*. Accessed: Apr. 13, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc1352p.pdf?ts=1618236228774&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC1352P
[45] *The Scyther Tool*. Accessed: Jan. 11, 2021. [Online]. Available: https://people.cispa.io/cas.cremers/scyther/
[46] The AVISPA Team. (Jun. 2006). *AVISPA V1.1 User Manual, Information Society Technologies Programme*. [Online]. Available: http://avispa-project.org
[47] *Texas Instruments: What's New in ZigBee 3.0*. Accessed: Aug. 19, 2020. [Online]. Available: https://www.ti.com/lit/an/swra615a/swra615a.pdf
[48] T. Genet, "A short SPAN+AVISPA tutorial," Ph.D. dissertation, IRISA, Rennes, France, 2015.
[49] D. Basin, S. Mödersheim, and L. Viganò, "An on-the-fly model-checker for security protocol analysis," in *Proc. Eur. Symp. Res. Comput. Secur.* Springer-Verlag, 2003, pp. 253–270.
[50] M. Turuani, "The CL-Atse protocol analyser," in *Proc. Int. Conf. Rewriting Techn. Appl.* Berlin, Germany: Springer, 2006, pp. 277–286.
[51] A. Armando, R. Carbone, and L. Compagna, "SATMC: A SAT-based model checker for security protocols, business processes, and security APIs," *Int. J. Softw. Tools Technol. Transf.*, vol. 18, no. 2, pp. 187–204, Apr. 2016.
[52] R. Küsters and T. Wilke, "Automata-based analysis of recursive cryptographic protocols," in *Proc. Annu. Symp. Theor. Aspects Comput. Sci.* Berlin, Germany: Springer-Verlag, 2004, pp. 382–393.
[53] D. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, Jan. 2012.

[54] J. Engel, J. Gerretzen, E. Szymańska, J. J. Jansen, G. Downey, L. Blanchet, and L. M. C. Buydens, "Breaking with trends in pre-processing?" *Trends Anal. Chem.*, vol. 50, pp. 96–106, Oct. 2013.
[55] R. E. Rhodes, M. D. Guerrero, L. M. Vanderloo, K. Barbeau, C. S. Birken, J.-P. Chaput, G. Faulkner, I. Janssen, S. Madigan, L. C. Mâsse, T.-L. McHugh, M. Perdew, K. Stone, J. Shelley, N. Spinks, K. A. Tamminen, J. R. Tomasone, H. Ward, F. Welsh, and M. S. Tremblay, "Development of a consensus statement on the role of the family in the physical activity, sedentary, and sleep behaviours of children and youth," *Int. J. Behav. Nutrition Phys. Activity*, vol. 17, no. 1, pp. 1–31, Dec. 2020.
[56] S. K. Goudos and G. Athanasiadou, "Application of an ensemble method to UAV power modeling for cellular communications," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2340–2344, Nov. 2019.
[57] U. U. Rehman, D. J. Chang, Y. Jung, U. Akhtar, M. A. Razzaq, and S. Lee, "Medical instructed real-time assistant for patient with glaucoma and diabetic conditions," *Appl. Sci.*, vol. 10, no. 7, p. 2216, Mar. 2020.
[58] R. K. Kodali and N. V. S. N. Sarma, "Energy efficient ECC encryption using ECDH," in *Emerging Research in Electronics, Computer Science and Technology*. New Delhi, India: Springer, 2014, pp. 471–478.
[59] Z. Vahdati, S. Yasin, A. Ghasempour, and M. Salehi, "Comparison of ECC and RSA algorithms in IoT devices," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 16, pp. 1–16, 2019.

UBAID UR REHMAN received the M.S. degree from the National University of Sciences and Technology, Pakistan, in 2015. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University, South Korea. He has three years of working experience in academia. His research interests include cloud computing security, fog computing, network security, activity recognition, and the Internet of Things.

SEONG-BAE PARK (Member, IEEE) received the B.S. degree in computer science from Korea Advanced Institute of Science and Technology, in 1994, and the M.S. degree in computer engineering and the Ph.D. degree in computer science and engineering from Seoul National University, in 1996 and 2002, respectively. He was a Professor of computer science and engineering with Kyungpook National University, from 2004 to 2017. In 2018, he joined Kyung Hee University, where he is currently a Full Professor of computer science and engineering. His research interests include machine learning, natural language processing, text mining, information extraction, and bio-informatics.

SUNGYOUNG LEE (Member, IEEE) received the B.S. degree from Korea University, Seoul, South Korea, and the M.S. and Ph.D. degrees in computer science from the Illinois Institute of Technology, Chicago, IL, USA, in 1987 and 1991, respectively. He was an Assistant Professor with the Department of Computer Science, Governors State University, University Park, IL, USA, from 1992 to 1993. He has been a Professor with the Department of Computer Engineering, Kyung Hee University, South Korea, since 1993, where he has been the Director of the Neo Medical Ubiquitous-Life Care Information Technology Research Center, since 2006. He is currently the Founding Director of the Ubiquitous Computing Laboratory. His current research interests include ubiquitous computing and applications, wireless *ad hoc* and sensor networks, context-aware middle-ware, sensor operating systems, real-time systems and embedded systems, and activity and emotion recognition. He is a member of ACM.