Efficient and Privacy-Aware Multi-party Classification Protocol for Human Activity Recognition

Zakaria Gheid, Yacine Challal, Xun Yi, Abdelouahid Derhab

Efficient and Privacy-Aware Multi-party Classification Protocol for Human Activity Recognition

Zakaria Gheid[a,*], Yacine Challal[a,b], Xun Yi[c], Abdelouahid Derhab[d]

[a] *Laboratoire de Méthodes de Conception des Systèmes, Ecole Nationale Supérieure d'Informatique, Algiers, Algeria*
[b] *Centre de Recherche sur l'Information Scientifique et Technique, Algiers, Algeria*
[c] *The School of Computer Science and IT, RMIT University, Melbourne, Victoria, Australia*
[d] *Center of Excellence in Information Assurance (COEIA), King Saud University, Riyadh, Saudi Arabia*

Abstract

Human activity recognition (HAR) is an important research field that relies on sensing technologies to enable many context-aware applications. Nevertheless, tracking personal signs to enable such applications has given rise to serious privacy issues, especially when using external activity recognition services. In this paper, we propose (Π-Knn): a privacy-preserving version of the K Nearest Neighbors (k-NN) classifier that is mainly built on (Π-CSP+): a novel cryptography-free private similarity evaluation protocol. As a sample application, we consider a medical monitoring system enhanced with a HAR process based on our privacy preserving classifier. The integration of the privacy preserving HAR aims to improve the accuracy of the clinical decision support. We conduct a standard security analysis to prove that our protocols provide a complete privacy protection against malicious adversaries. We perform a comparative performance evaluation through several experiments while using real HAR system parameters. Experimental evaluations show that our protocol (Π-CSP+) incurs a low increasing overhead (37% in Online classification and 50% in Offline classification) compared to PCSC, representative state-of-the art protocol, which incurs 3600% and 4800% in online and offline classification respectively. Besides, Π-CSP+ provides a stable and efficient response time (W=0.0x m.seconds) for both short and long duration activities while serving up to 1000 clients. Comparative results confirm the computational efficiency of our protocol against a competitive state-of-the-art protocol.

Keywords: Human Activity Recognition, k-NN Classification, Multi-Party Computation, Privacy Preserving.

1. INTRODUCTION

Data mining methods are gaining an increasing attention because of the wide proliferation of knowledge-based applications. Analyzing data from wireless and sensor networks has enabled developing new services, such as Human Activity Recognition (HAR). HAR consists of tracking environmental and personal sensed signs, then, analyzing them to provide accurate information about persons' daily activities. Nevertheless, the collection and analysis of personal private data, such as GPS location, raises concerns about users' privacy, especially when the analysis is performed through external service providers. External recognition aims to reduce the cost of computation and storage accrued by client devices. Additionally, it aims ensuring a high accuracy level in recognition results, which are built upon big data stores of activity patterns.

To face such a concern, several Privacy-Preserving Data Mining (PPDM) methods have been proposed. These include classification, clustering and other data mining tasks [1]. PPDM methods protect the privacy

*Corresponding author
Email addresses: z_gheid@esi.dz (Zakaria Gheid), y_challal@esi.dz (Yacine Challal), xun.yi@rmit.edu.au (Xun Yi), abderhab@ksu.edu.sa (Abdelouahid Derhab)

by changing or deleting sensitive data before analysis [2]. This approach is based on a trade-off between accuracy and privacy [3]. Other approaches employed cryptographic techniques to provide a high privacy protection level, but, they are computationally very expensive [4].

From another side, privacy-preserving HAR may provide useful information that enhances context-aware aspects in several applications, like e-healthcare monitoring systems. CodeBlue [5], AlarmNet [6] and some other popular medical monitoring systems [7, 8] have been proposed and focused on addressing power, security and computational resource constraints [9]. Yet, they have some shortages in tracking information about patients' physical activities. Such information is useful to avoid wrong diagnosis and treatment when vital sensed signs are jammed, errored or modified. To shed light on this, studies of information needs by clinicians show that in about 81 percent of ambulatory diagnosis, physicians are missing critical information [10]. Other studies report that about 18 percent of medical errors may be due to insufficient availability of patient information [11]. Thus, acquiring a complete picture of patient state will reduce medical errors and may drive for a broad adoption of e-healthcare monitoring systems for the clinical decision support (CDS) task.

In this paper, we propose a novel privacy-preserving k-NN classification version, which aims to address privacy and efficiency concerns when using external services for human activity recognition. As an application, we propose a framework that combines the human activity recognition (HAR) process with the clinical decision support (CDS) process. This may enhance accuracy in medical decision while protecting patients' privacy.

We summarize the contributions of this work in the following items

- We build a novel privacy-preserving version of k-NN, named (Π-Knn), and we use it for the classification task, which is applied according to external activity patterns.
- We propose (Π-CSP+), a novel privacy-preserving and efficient cosine similarity protocol, which is the main core of (Π-Knn). It aims to securely assess similarity between HAR sensed data and external activity patterns. Π-CSP+ is based on simple arithmetic operations to avoid computation overheads induced by cryptographic techniques.
- As an application of the HAR system, we propose SimilCare, A novel medical monitoring framework that embeds information about patients' activities within a clinical knowledge database while using our proposed Π-Knn protocol. SimilCare aims to cover shortage of existing healthcare monitoring systems in tracking information about patients' activities, while ensuring their privacy.
- We present a security analysis of our proposed protocols (Π-CSP+ and Π-Knn) using a standard security proof [12], which has revealed a complete privacy protection. In addition, we perform simulations through different experiments while using real HAR system parameters. The computation performances are highly efficient compared to the most efficient protocol found in the literature [4].

The remainder of this paper is organized as follows. In section 2, we provide a literature survey of related works and we discuss them. Section 3 presents preliminaries and building blocks used for designing our protocols. Next, we devote section 4 to present our privacy-preserving protocols, besides their integration in the proposed SimilCare framework. Then, we evaluate the privacy protection and the performance level in section 5 and 6 respectively. We end-up this work with our final conclusions in section 7.

2. RELATED WORK

Several existing HAR systems have not considered protecting users' privacy during the recognition and classification phase. In this section, we review recent works in HAR field. Besides, we give a review on privacy-preserving k-NN classification, and privacy-preserving similarity evaluation, which is the main privacy-related computation within k-NN protocol.

2.1. Human activity recognition (HAR)

Najafi et al. [13] proposed an activity recognition system based on Kinematic sensors. This system aims to monitor elderly people in their daily lives. Authors have focused on accuracy in activities detection but gave no privacy preserving measurements. Hou et al. [14] proposed PAS: an open architecture that exploits off-the-shelf technologies to assist elderlies. PAS incorporated mechanisms to secure data storage and communication, however, there is no privacy protection of sensed data during recognition process. Jiang et al. [15] proposed CareNet: a system prototype for remote physical activity monitoring in healthcare application. CareNet provides secure communication while there are no privacy protection measurements regarding data storage and analysis. Lau et al. [16] introduced CARMA: A Context Aware Remote Monitoring Assistant application, which enables activity recognition for patients using non-obtrusive devices. CARMA aims to assist clinicians to obtain implicit information regarding the patients' context. Authors focused on the recognition accuracy and did not provide any privacy-preserving specification. Evani et al. [17] proposed a patient activity monitoring system using wearable flex sensors. Their system recognizes sitting, standing and walking activities as well as inactivity. Regarding data privacy, authors did not specify any protection measurement. Recently, De et al. [18] introduced a fine-grained activity recognition system using multi-modal wearable sensors. Authors highlighted the need for detecting complex activities in critical healthcare applications. The proposed system could recognize 19 in-home activities without using video recording that induces direct privacy concerns. Although the use of only wearable devices, some sensed data, such as GPS location, could breach the privacy of the monitored persons.

2.2. Privacy-preserving k-NN

Privacy issue in k-NN classification has been largely tackled in literature works. Xiong et al. [19] presented a framework including multi-round algorithms for mining horizontally partitioned databases using a privacy preserving k-NN classifier. Their approach made a trade-off between accuracy, efficiency and privacy. Recently, Mynavathi et al. [20] used Gaussian noise to build a novel secure k-NN classifier, which aims to provide better secured data mining result with minimum information loss. Such approach based on data perturbation methods balanced the privacy preservation and the accuracy of data mining. Other approaches [21, 22, 23] employed cryptographic techniques, such as homomorphic encryptions, to build privacy-preserving k-NN classifiers over encrypted data.

In this work, we propose a secure and cryptography-free k-NN classifier that provides a complete privacy protection without information loss. We build this classifier on a novel privacy-preserving cosine similarity evaluation protocol free from cryptographic operations. Secure similarity evaluation is the main core in building secure k-NN classification.

2.3. Privacy-preserving cosine similarity

Several works have already been proposed to secure the cosine similarity metric that we use in this work. This is equivalent to secure the scalar product evaluation (section 3.3). Vaidya and Clifton [24] proposed a privacy preserving scalar product protocol, which was based on algebraic operations to scale well to large data sets. Nonetheless, B. Goethals et al. [25] identified some attacks against this protocol with binary values and proposed another secure protocol based on Homomorphic encryption. Hiroaki Kikuchi et al. [26] proposed a secure similarity evaluation using the cosine correlation and the Euclidean distance by implementing two Homomorphic encryption-based protocols. Yang et al. [27] proposed an ElGamel encryption-based protocol for secure cosine similarity computation which resists to malicious adversaries. LU et al. [4] proposed a privacy-preserving cosine similarity protocol for big data analytics. Authors argued that encryption-based methods are not adequate for large scale data analysis. Thus, they built their proposal on simple arithmetic operations without any cryptographic scheme. Recently, Huang et al. [28] proposed a secure scalar product protocol for wireless sensor networks. Their protocol was based on Homomorphic encryption to protect privacy under semi-honest model of adversaries. Zhu et al. [29] proposed a secure two-party scalar product protocol. Authors affirmed having no extra communication overheads compared to the scalar product computation without privacy-protection.

To summarize, we can classify existing privacy-preserving similarity evaluation works into two categories: a) methods that are based on cryptographic schemes [25, 30, 31, 32, 26, 27, 28, 29], such as Homomorphic encryption. These methods are trading security and performance. Even though they are providing a high privacy protection, they are inducing unaffordable overhead in computational time, which is not suitable for real-time services. b) The second class involves cryptography-free methods [24, 31, 32, 33, 4], which use light-weight arithmetic transformations to protect private data. The latter class ensures providing services with high computational efficiency at the expense of privacy. Almost all methods that fall in this class operate under some condition, such as a specific data type (integer or binary).

3. BACKGROUND

3.1. Human activity recognition (HAR)

Human activity recognition (HAR) is the field that aims to provide accurate information on people's activities. The general structure of a HAR system involves three main phases, as shown in Fig. 1.

- In **data collection** phase, the sensors' raw data are communicated to the data collection node. Sensors are attached to different locations on the body or placed in the environment. The raw data are sampled in a multivariate time series (s_j^i) depending on sensors frequencies, where j corresponds to a sensed attribute and i is the sample number.
- During **feature extraction** phase, the large raw data are transformed into a reduced representation set of features that are more discriminative for the activities at hand. Given a set of time windows (w_i) equal in size, each of which involves a set of time-series from the sensed raw data. Then, a feature vector is created in each time window and passed to classification phase.
- During **classification** phase, different algorithmic methods (nearest neighbor, neural networks,...) could be used to classify feature vectors got from the precedent phase according to activity patterns.

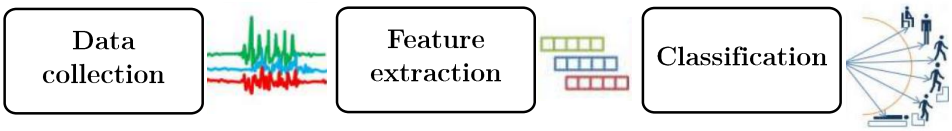


Figure 1: General structure of a HAR process

Let HARP denote the problem of recognizing human activities from sensors' raw data, which is formally defined bellow [34].

Definition 1 (HARP)**.** *Let $A = \{a_1, ..., a_k\}$ be a set of activities labels and $W = \{w_0, ..., w_n\}$ be a set of n time windows equal in size. We assume each w_i includes a set of time series $S_i = \{s_0^i, ..., s_m^i\}$ from the m measured attributes. Then, the HAR problem returns to find a mapping function $f : S_i \mapsto A$ such that $f(S_i)$ is as similar as possible to the activity performed in w_i*

In this work, we focus on classification phase. We will use the k-NN classifier as a mapping function (f) to solve the HARP.

3.2. k-NN classification

Classifiers are machine learning tools used for solving the HAR problem [34]. In this work we leverage the k-nearest neighbors (k-NN) classifier, which is one of the most used classification methods [35]. Given x a new object, we compute its k nearest neighbors from a set of already classified objects according to a distance/similarity metric, then, we assign x to the most represented class in the set of the k nearest neighbors. Let $D = \{(x_1, y_1), ..., (x_p, y_p)\}$ be a set of p labelled objects where x_i and y_i correspond to the

object data and the object class respectively. Let z be a new object that we want to get its class denoted y_z. Given a function g, we define the set of points x for which g reaches its largest value as

$$\underset{x}{\operatorname{argmax}}\, g(x) = \{x | \forall y : g(x) \geq g(y)\}$$

Algorithm 1 bellow presents the k-NN classification process.

Algorithm 1: k-NN classification

Input : D, z and k, where: $D = \{(x_1, y_1), ..., (x_p, y_p)\}$, z is the new object and $0 < k \leq p$.
Output: y_z, the class label of z.
1: Compute $\mathrm{d}(z, x_i)$, the distance/similarity between z and every object in D.
2: Select $D_z \subseteq D$, the set of k closest objects to z.
3: $y_z = \underset{c}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} I(c = y_i)$, where the identity function I converts $true$ and $false$ values to 1 and 0 respectively.

In this work, we leverage the use of cosine similarity metric within the k-NN process (instruction 1, Algorithm 1). This metric yields good results when evaluated in k-NN context [36].

3.3. Cosine similarity

Cosine similarity is a statistical metric used among others to find nearest neighbors by k-NN classifiers [36]. Given two vectors of numerical attributes $\vec{a} = (a_1, ..., a_n)$ and $\vec{b} = (b_1, ..., b_n)$, we get the cosine metric value by

$$\cos(\vec{a}, \vec{b}) = \frac{(\vec{a} \cdot \vec{b})}{\|\vec{a}\| \|\vec{b}\|} \tag{1}$$

where $(\vec{a} \cdot \vec{b})$ is the scalar product obtained as

$$(\vec{a} \cdot \vec{b}) = \sum_{i=1}^{n} (a_i \times b_i) \tag{2}$$

and $\|\vec{a}\|$ (resp. $\|\vec{b}\|$) is the Euclidean norm that could be shortened when dealing with normalized vectors. Assume $\hat{a} = (\vec{a}/\|\vec{a}\|)$ and $\hat{b} = (\vec{b}/\|\vec{b}\|)$ the normalized representation of $\vec{a}$ and $\vec{b}$ respectively. Then

$$\cos(\vec{a}, \vec{b}) = (\hat{a} \cdot \hat{b}) \tag{3}$$

In our work, we will consider securing the computation of this metric between two different parties. This aims to build a secure k-NN classifer that we name Π-Knn and we embed within the proposed monitoring framework explored in next section.

4. Π-Knn: A PRIVACY-PRESERVING AND EFFICIENT k-NN CLASSIFICATION PROTOCOL FOR HUMAN ACTIVITY RECOGNITION

In this section, we present an efficient and privacy-preserving k-NN algorithm called Π-Knn. We build this protocol on a privacy-preserving Cosine Similarity Protocol that we call Π-CSP+. Next, we integrate these protocols in SimilCare, a novel proposed medical monitoring framework.

4.1. Motivation

Despite numerous studies and projects already developed in HAR field (section 2.1), the adoption of such important application in real-life settings is still lacking. Two relating issues are to be expected: **user privacy** and **service efficiency**. In fact, HAR systems raised several privacy concerns surrounding mining personal tracked signs. This is highly true in service-oriented HAR, where personal signs are sensed locally, then, exported to be classified according to activity patterns held by external servers. Facing this, privacy-preserving classifiers have been proposed relying heavily on cryptographic schemes, such as homomorphic encryptions (section 2.2). This, raised a novel issue related to efficiency in computational time [4]. It is therefore the vision of this work to propose both a private and efficient k-NN classifier that we named Π-Knn. Our classifier is based on a novel private similarity protocol named Π-CSP+, and built on efficient matrix algebra without involving any cryptographic scheme.

4.2. k-NN privacy issue

Let us consider a HAR system denoted S_1 that performs the classification phase (section 3.1) using the k-NN process (section 3.2) according to activity patterns held by an external service provider denoted S_2. In this case, S_1 and S_2 need to collaborate in computing the distance/similarity between each recorded activity and the class patterns (instruction 1, Algorithm 1). Let $\vec{z} = (z_1, ..., z_n)$ denote a feature vector of a new activity recorded by S_1 and let $\vec{p_j} = (p_{(1,j)}, ..., p_{(n,j)})$ denote the pattern of the class j. As we leverage the use of the cosine similarity metric (section 3.3) because of its good accuracy level in k-NN contexts [36], we formalize the collaboration between S_1 and S_2 as follows

$$cos(\vec{z}, \vec{p_j}) = (\hat{z} \cdot \hat{p}_j) = \sum_{i=1}^{n} (\hat{z}_i \times \hat{p}_{(i,j)}) \tag{4}$$

Where $\hat{z}$ and $\hat{p}_j$ denote the normalized representation of $\vec{z}$ and $\vec{p_j}$ respectively. Such a computation needs to disclose attributes of one site to the other, which is considered as a privacy breach for both S_1 or S_2 when it comes to private data. This may be a common situation for HAR distributed applications where the client monitoring system makes use of external activity patterns. External analytics aim to shorten the learning phase required by classifier tools [34] and should improve the response time owing to the high computational capacity of service providers. Thus, in order to tackle this privacy issue we propose Π-CSP+: a privacy-preserving and efficient cosine similarity protocol.

4.3. Π-CSP+: a Privacy-preserving and efficient cosine similarity protocol

In this subsection, we present the Π-CSP+ that aims to ensure the privacy protection of the similarity evaluation task. This protocol measures the similarity of the set of maximum available individuals in the minimum communication steps. We adopt a cryptography-free communication scheme in order to provide a high efficient service. This scheme is based on an attribute-independent noise, which enhances its scalability contrary to existing similar approaches [4] that generate a scalar noise for each object attribute.

Let us consider two different sites S_1 and S_2 having respectively $A = \{\hat{a}_1, ..., \hat{a}_p\}$ and $B = \{\hat{b}_1, ..., \hat{b}_v\}$ sets of objects that involve private sensitive data. We assume for $(1 \leq i \leq p)$ and $(1 \leq j \leq v)$: $\hat{a}_i$ and $\hat{b}_j \in \mathbb{R}^n$, they have the same structure and they result from a normalization process. Under these assumptions the cosine similarity between the original objects of A and B will be shortened to the scalar product of the correspondent normalized objects (section 3.3). We define $M_R[p \times p]$, $M_A[p \times n]$ and $M_B[n \times v]$ as matrix tools used during the scalar product process, where M_R is a random noise, M_A involves the p data objects of A and M_B includes the v data objects of B. We build M_A from data objects put as rows and we build M_B from data objects put as columns. Assume M_R is an invertible matrix, $(p, n, v) \in N^{3*}$ such as: $(1 < p < n$ and $0 < v < p)$. Implementation of Π-CSP+ is detailed in Algorithm 2.

Note 1. *in Π-CSP+ we consider the data normalization process explicitly. This may help to avoid any confusion between the cosine similarity of the original vectors and the cosine similarity of the normalized ones.*

Algorithm 2: Π-CSP+, a Privacy-preserving and Efficient Cosine Similarity Protocol

Input : $A_{origin} = \{\vec{a_1}, ..., \vec{a_p}\}$ S_1 data objects
$B_{origin} = \{\vec{b_1}, ..., \vec{b_v}\}$ S_2 data objects

Output: (For S_1 only) $M_{AB}[p \times v]$ containing the cosine similarity results, where $M_{AB}[i, j] = cos(\vec{a_i}, \vec{b_j})$

Preprocessing: S_1 and S_2 compute respectively $A = \{\hat{a}_1, ..., \hat{a}_p\}$ and $B = \{\hat{b}_1, ..., \hat{b}_v\}$ the sets of normalized objects from A_{origin} and B_{origin}.

Step 1 by S_1

1: Generates a random invertible matrix $M_R[p \times p]$
2: Puts A's elements as rows in a matrix $M_A[p \times n]$
3: Performs $(M_R \times M_A)$ and sends the result matrix (M_{RA}) to S_2

Step 2 by S_2

4: Puts B's elements as columns in a matrix $M_B[n \times v]$
5: Performs $(M_{RA} \times M_B)$ and sends back the result matrix (M_{RAB}) to S_1

Step 3 by S_1

6: Performs $(M_R^{-1} \times M_{RAB}) = (M_A \times M_B) = M_{AB}$ which is the searched cosine similarity matrix.

4.4. Π-Knn description

Based on Π-CSP+ presented above we build a privacy-preserving version of the k-NN process (Algorithm 1) and we call it Π-Knn. In order to adapt the similarity evaluation task of the k-NN algorithm to Π-CSP+ we divide each time window w_i, which is considered as a time unit for one classification (Definition 1), into v sub-windows. Thereby, in each classification we will consider v recorded activities each of which has a separate extracted feature vector.

Like in Algorithm 1, let us assume $D = \{(x_1, y_1), ..., (x_p, y_p)\}$ a set of p patterns held by a service provider denoted SP, such as for $(1 \leq i \leq p)$: $x_i \in \mathbb{R}^n$ and has the correspondent y_i class label. Assume $Z =< z_1, ..., z_v >$ a set of v observations in a SimilCare client system denoted SC, such as for $(1 \leq j \leq v)$: z_j is the feature vector extracted from the observation j and has $y_{(z,j)}$ as activity class that we are searching for. Assume for $(1 \leq j \leq v)$: $z_j \in \mathbb{R}^n$ and it has the same structure of x_i for $(1 \leq i \leq p)$. Π-Knn classification of the v observations is detailed in Algorithm 3.

Algorithm 3: Π-Knn, a Privacy-preserving and Efficient k-NN classification protocol

Input : $D = \{(x_1, y_1), ..., (x_p, y_p)\}$: $x_{i,(1 \leq i \leq p)} \in \mathbb{R}^n$
$Z =< z_1, ..., z_v >$: $z_{j,(1 \leq j \leq v)} \in \mathbb{R}^n$
$1 < p < n$
$0 < v < p$
$0 < k \leq p$

Output: $< y_{(z,1)}, ..., y_{(z,v)} >$, the correspondent class label of each observation within Z

Step 1 by $(SP \cup SC)$

1: Compute Π-CSP+$(\{x_1, ..., x_p\}, \{z_1, ..., z_v\})$, the cosine similarity matrix using the Π-CSP+ protocol.

Step 2 by SP

2: **for** $(j = 1; j <= v; j++)$ **do**
3: Select $D_{(z,j)} \subseteq D$, the set of k patterns having the highest similarity rate in the column j of the cosine similarity matrix got from 1.
4: $y_{(z,j)} = \underset{c}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_{(z,j)}} I(c = y_i)$, where the identity function I converts $true$ and $false$ values to 1 and 0 respectively.
5: **end for**
6: **return** $< y_{(z,1)}, ..., y_{(z,v)} >$ to SC

4.5. SimilCare: an activity-aware medical monitoring framework

We designed SimilCare by integrating the Human Activity Recognition (HAR) process within a Clinincal Decision Support process. We leverage the use of a standard design for both components in order to better support the adoption of this framework by existing systems. We sketch the whole framework design in Fig. 2 and the privacy-related components in Fig. 3. In what follows, we describe each SimilCare component.

- **Monitored patient**: It represents the patient monitored by SimilCare. He/she interacts with the monitoring process of SimilCare across wearable, portable and implantable sensors, which track personal and environmental signs to feed both HAR and CDS components.
- **HAR component**: It performs three main tasks (Fig. 1). The data collection task is carried out by different sensors while considering only environmental and acceleration signals, which are stored in the HAR data server. We avoid using physiological signs in HAR process since they do not provide useful information in activity recognition [34]. Nevertheless, because they are available for the CDS component they could be used whenever needed by a HAR system. The HAR service extracts features from the HAR data server, composes feature vectors and contacts the patterns service provider for a classification task. The classification is performed using our proposed privacy-preserving protocol (Π-Knn & Π-CSP+). In the end, the HAR service stores the activity recognition report within the knowledge data server in order to be used by the CDS component.
- **CDS component**: It consists of a clinical decision support system with a standard design and an extended output node (alarms, calls and SMS). It includes an input data node, which is the vital data server that stores physiological data signs, a knowledge database stored within the knowledge data server and containing rules as well as activities' reports and an inference engine that consists of the CDS service. The outputs of this component are handled by I/O service in order to display reports through HMIs, trigger alarms, make emergency calls or send messages. The knowledge database activities reports are affected by the HAR component and the rules are added across the user HMI, handled by I/O service before being stored into the knowledge data server.
- **Privacy-preserving and efficient classification**: This is a multi-party component that involves the proposed privacy-preserving multi-party classification protocol (Π-Knn & Π-CSP+) as shown in Fig. 3. The client part is implemented on the HAR component and the server part is on the pattern service provider side.
- **Pattern service provider**: This is the external part of SimilCare. It consists of any external service that will provide a set of activity patterns to enable the classification of extracted feature vectors. It represents the server side during the execution of the multi-party classification protocol.
- **Users**: This part represents all users that interact with the framework. This includes medical staff, system administrator as well as the monitored patient. Interactions of users with SimilCare may be for reading notifications and medical reports or for entering decision rules and system configurations.

Note 2. *Recall that we designed SimilCare as a standard infrastructure paradigm that would integrate any HAR system with a CDS system to provide a secure activity-aware medical monitoring. Therefore, we do not specify any secure protocol that should be implemented for internal data storage or intern communications, which is out of the scope of this work. The main purpose of this work is to provide a privacy-preserving classification protocol (*Π*-Knn and* Π*-CSP+) that we embed within SimilCare to provide activities information about monitored patients while preserving their privacy.*

4.6. Π-Knn within SimilCare: How it runs ?

For generalization purpose we avoided to distribute sites' roles (client and server) when presenting Π-CSP+ (Algorithm 2). Thereby, we give it more adaptability for several contexts. Without loss of generality, for SimilCare application we consider substituting respectively S_1 and S_2 presented in Π-CSP+ (Algorithm

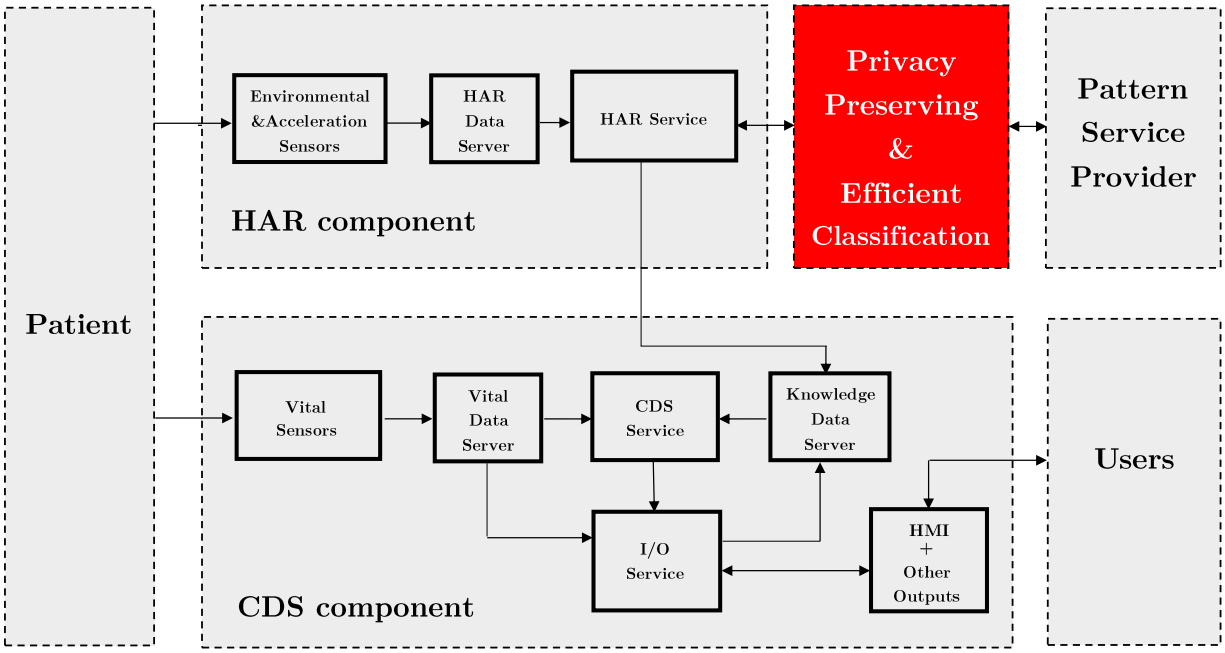


Figure 2: The design model of SimilCare framework

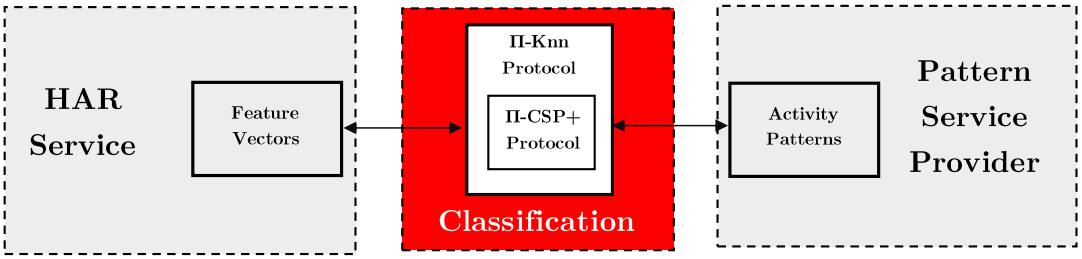


Figure 3: Privacy-related components in SimilCare framework

2) by the patterns service provider and the SimilCare HAR service (the HAR classification module) denoted respectively SP and SC within the Π-Knn protocol (Algorithm 3). In Fig. 4, we sketch a running process of Π-Knn and Π-CSP+ during a SimilCare classification sequence.

5. SECURITY ANALYSIS

In this section, we provide a security analysis of our proposal according to the real/ideal simulation paradigm [12, 37]. We stress that such a proof provides very strong security guarantees [37].

Note 3. *Notice for clarification that real/ideal simulation given in this section has no relation with simulation made for the performance evaluation in the next section.*

5.1. Definitions & Notations

Definition 2 (Two-party computation)**.** *Assume P_1 and P_2 two parties having respectively v_1 and v_2 private data and want to jointly get the result of the application of a public function f at the point (v_1, v_2). Thus,*

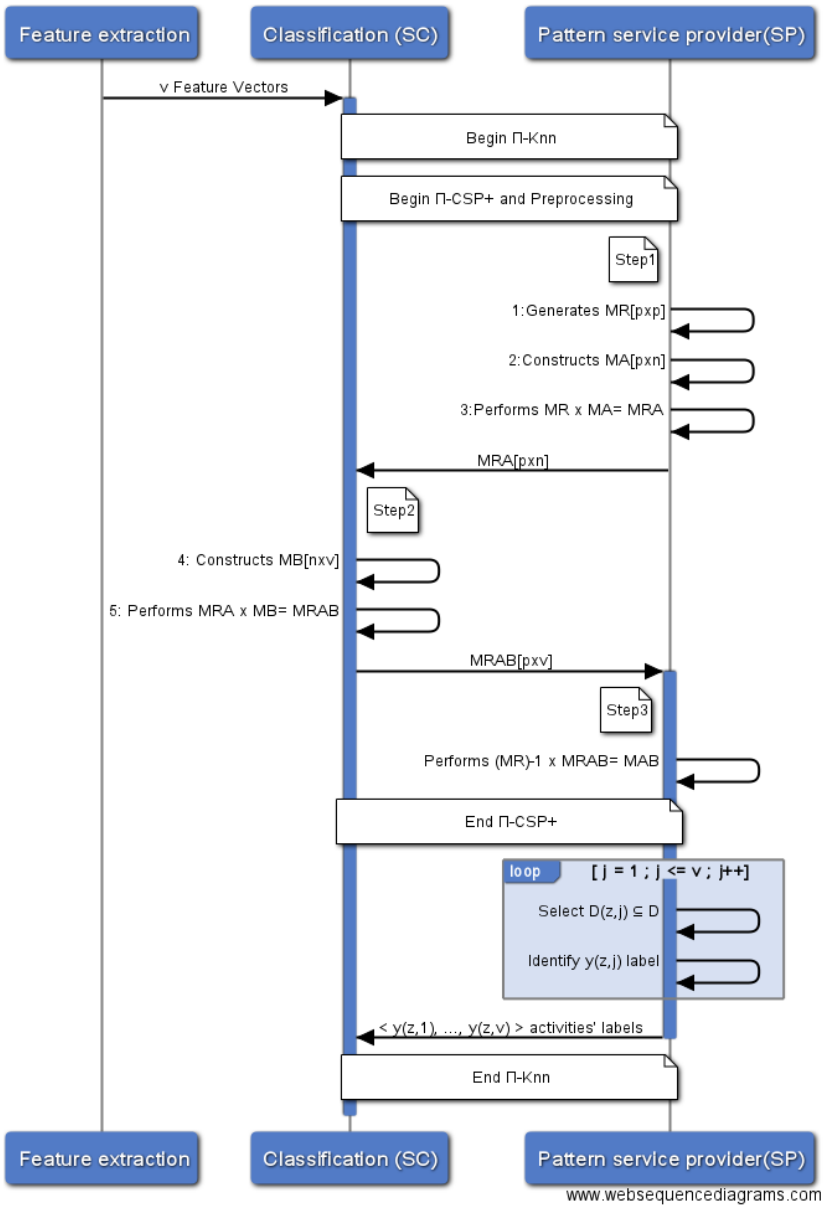


Figure 4: Π-Knn and Π-CSP+ running process in a SimilCare classification sequence.

$f(v_1, v_2)$ *is called two-party computation [37].*

Definition 3 (Adversary models)**.** *The allowed behavior of corrupted parties that participate to a two-party computation protocol can be classified according to the adversary's model [37].*

a) ***Passive adversary.*** *Also called semi-honest adversary, which is supposed following the protocol specifications yet it is allowed to analyze all information gathered by corrupted parties during the execution of the protocol.*

b) ***Active adversary.*** *Also called malicious adversary, which allows corrupted parties to randomly deviate from the protocol specifications. The two common behaviors of such an adversary in a two-party computation are a) aborting the protocol untimely or b) injecting fake inputs.*

Notation 1. *Let* Π *denote a two-party protocol executed by* P_1 *and* P_2 *in order to evaluate a function* f *on the join of their private inputs* M_A *and* M_B *respectively. Let param denote the set of security parameters used during the execution of* Π.

5.2. Real/Ideal paradigm

In what follows we introduce the real/ideal paradigm used for the security proof. Let Π, *param*, M_A and M_B be as defined above.

- In a **real model execution** we consider the presence of a real adversary denoted A, which corrupts one of the two parties and behaves according to an adversarial model (Definition 3). At the end of the execution, uncorrupted party outputs which was specified in the protocol and the corrupted party outputs any random function of A's view, which involves its inputs, its outputs and the messages it gets during the execution[1]. Let $\mathbf{RE_{\Pi,A}(z,param,M_A,M_B)}$ denote the global output of a real-life execution of Π on inputs M_A, M_B, security parameters param and auxiliary input z, which involves additional information that A may have (such as the nature of inputs and the expression of f). Let $\mathbf{RE_{\Pi,A,i}(z,param,M_A,M_B)}$ denote the output of the party P_i ($i \in \{1,2\}$) in the same execution of Π. Then

$$\mathrm{RE}_{\Pi,A}(z, param, M_A, M_B) = \mathrm{RE}_{\Pi,A,1}(z, param, M_A, M_B) \cup \mathrm{RE}_{\Pi,A,2}(z, param, M_A, M_B)$$

- In an **ideal model execution** we assume the presence of an incorruptible trusted party denoted T that receives parties' inputs in aim to evaluate the function f, then delivers the expected value to each party. We assume the presence of an ideal adversary denoted S that handles inputs of the corrupted party and behaves according to an adversarial model before sending them to T. By the end, the uncorrupted party outputs what was received from T while the corrupted party outputs a random function of S's view during the execution (which consists of the corrupted party's input, S's eventual changed inputs and values received from T). Let $\mathbf{ID_{f,S}(z,param,M_A,M_B)}$ denote the global output of an ideal execution of f applied on inputs M_A, M_B, security parameters param and auxiliary input z, which involves additional information that S may have. Let $\mathbf{ID_{f,S,i}(z,param,M_A,M_B)}$ denote the output of the party P_i ($i \in \{1,2\}$) in the same execution. Then

$$\mathrm{ID}_{f,S}(z, param, M_A, M_B) = \mathrm{ID}_{f,S,1}(z, param, M_A, M_B) \cup \mathrm{ID}_{f,S,2}(z, param, M_A, M_B)$$

5.3. Security definition

Let Π, *param*, M_A and M_B be as above. Let $\stackrel{d}{\equiv}$ denote the distribution equality.

Definition 4 (Secure two-party protocol)**.** *We consider* Π *as a secure two-party protocol if for any real adversary A that behaves according to some adversary model (Definition 3) while attacking the protocol* Π*, there exists an ideal adversary S having the same adversary model such that for a fixed security parameter param, we have on any inputs* M_A*,* M_B *and auxiliary input z*

$$\{\mathrm{ID}_{f,S}(z, param, M_A, M_B)\} \stackrel{d}{\equiv} \{\mathrm{RE}_{\Pi,A}(z, param, M_A, M_B)\}$$

By this global security definition (outputs of corrupted and uncorrupted parties together) we ensure the intertwined [12] security requirements that are privacy and correctness. To clarify this, let c and u denote respectively the index of corrupted and the uncorrupted party. Then:

- We protect the **privacy** so that any information output by c in the real execution could be output in the ideal one and this is by requiring

$$\{\mathrm{ID}_{f,S,c}(z, param, M_A, M_B)\} \stackrel{d}{\equiv} \{\mathrm{RE}_{\Pi,A,c}(z, param, M_A, M_B)\}$$

[1] In other equivalent formalizations, the adversary outputs its view and the corrupted party has no output. We leverage our formalization for its simplicity.

- We guarantee the **correctness** so that any information output by u in the real execution could be output in the ideal one and this is by requiring

$$\{\mathrm{ID}_{f,S,u}(z, param, M_A, M_B)\} \stackrel{d}{\equiv} \{\mathrm{RE}_{\Pi,A,u}(z, param, M_A, M_B)\}$$

5.4. Security proof

In this subsection, we will prove the security of the Π-CSP+ through the real/ideal simulation relying on definitions and notations given above. Next, we deduce the security of the Π-Knn protocol.

Theorem 1. *The* Π*-CSP+ presented by Algorithm 2 is a secure two-party protocol in the presence of an active adversary.*

Proof. Let Π denote the Π-CSP+ and let *param* be the set of security parameters defined as *param*= $\{p, n, v\}$ where $(1 < p < n)$ and $(0 < v < p)$. Assume S_1 and S_2 are the two participating parties and M_A and M_B are their inputs respectively. Let z denote an auxiliary input that each party may have and that involves information about Π-CSP+. In this proof we will consider separately the case where S_2 is corrupted and the case where S_1 is corrupted:

- **Case 1.** If S_2 is corrupted then it can inject fake inputs (M_B) (however, aborting the protocol untimely is not possible for this case because even the real execution will not run). In this case, S receives the fake M_B from S_2 and just sends it to T, thereby completing the simulation. By the end, the output of S_2 in the ideal and real execution are as follows

$$\begin{aligned} \mathrm{ID}_{f,S,2}(z, param, M_A, M_B) &= \{M_B\} \\ \mathrm{RE}_{\Pi,A,2}(z, param, M_A, M_B) &= \{M_B, M_{RA}\} \end{aligned} \tag{5}$$

  Nevertheless, relying on security parameters defined in *param*, M_{RA} will contain $((p \times p) + (p \times n))$ unknowns opposite to $(p \times n)$ equations, thus M_{RA} will not involve any information for S_2 and can be considered as a random noise. thus, (5) $\Rightarrow$

$$\{\mathrm{ID}_{f,S,2}(z, param, M_A, M_B)\} \stackrel{d}{\equiv} \{\mathrm{RE}_{\Pi,A,2}(z, param, M_A, M_B)\} \tag{6}$$

  On the other hand, the output of S_1 are as follows

$$\begin{aligned} \mathrm{ID}_{f,S,1}(z, param, M_A, M_B) &= \{M_A, M_{AB}\} \\ \mathrm{RE}_{\Pi,A,1}(z, param, M_A, M_B) &= \{M_A, M_{AB}, M_{RAB}\} \end{aligned} \tag{7}$$

  But, since we have $(p < n)$ defined in *param*, M_{RAB} will involve $(p \times v)$ equations and $(n \times v)$ unknowns, so, it can not reveal any information for S_1. Hence, M_{RAB} is considered as a random noise. Thus, (7)$\Rightarrow$

$$\{\mathrm{ID}_{f,S,1}(z, param, M_A, M_B)\} \stackrel{d}{\equiv} \{\mathrm{RE}_{\Pi,A,1}(z, param, M_A, M_B)\} \tag{8}$$

  Through (6) and (8) we have proved that when S_2 is corrupted by an active adversary, Π-CSP+ is a secure two-party protocol that correctly emulates an ideal process and verifies the definition 4. i.e.

$$\{\mathrm{ID}_{f,S}(z, param, M_A, M_B)\} \stackrel{d}{\equiv} \{\mathrm{RE}_{\Pi,A}(z, param, M_A, M_B)\} \tag{9}$$

- **Case 2.** If S_1 is corrupted then it can inject fake inputs (M_A) or abort the protocol in step 2. But, since S_2 does not require any output, the abort of S_1 will have no effect. Let us consider the case where S_1 sends a fake M_A. In this case, S will receive M_A from S_1 and just sends it to T in order to complete the simulation. By the end, S_2 and S_1 will output respectively (5) and (7). Like in case 1, M_{RA} and M_{RAB} could be seen as a random noise when considering security parameters defined in *param*. Thus, ((5) and (7)) $\Rightarrow$ ((6) and (8)) from which we can deduce that when S_1 is actively corrupted, Π-CSP+ is a secure two-party protocol that correctly emulates an ideal process and verifies definition 4.

□

Note 4. *We stress that we consider S_1 and S_2 changing their inputs (M_A and M_B respectively) in each execution of* Π-*CSP+. Thereby, additional data (M_{RAB} and M_{RA}) that we considered as a random noise for one execution will be likewise useless and do not involve any information for the next executions. Hence, we ensure the secure re-execution of* Π-*CSP+ for t ($t > 0$) times by the same parties. This assumption is perfectly valid in SimilCare context where the HAR service (client) would participate every time with feature vectors extracted from real-time sensed signs. As for the other side, HAR service will have no way to know if the patterns provider (server) have reordered its inputs or changed them by some novel extracted patterns.*

Corollary 1. *The* Π-*Knn protocol presented by Algorithm 3 is a secure two-party protocol in the presence of an active adversary.*

Proof. As the call to Π-CSP+ is the only multi-party task within Π-Knn, the proof of corollary 1 relies heavily on theorem 1 proved above. □

5.5. Limitation: What could Π-CSP+ disclose ?

In the case S_1 is corrupted, it can inject a fake M_A having the form

$$M_A = \begin{bmatrix} x_{11} & 0 & 0 & \dots & 0 \\ 0 & x_{22} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & x_{pp} & 0 \end{bmatrix}$$

where $(x_{11}, ..., x_{pp}) \in \mathbb{R}^{p*}$. This should disclose the p first rows of M_B. We notice the same behavior in all other scalar product protocols [24, 33, 4] and we stress that it is the expected final result of the scalar product, which would be obtained even in the ideal execution. This respects well the security definition in the real/ideal paradigm (Definition 4). Although, we can avoid such a disclosure risk in some contexts where accuracy of the inputs is crucial. For instance, in SimilCare context the pattern service provider should provide valid patterns to the HAR service, otherwise, the CDS service may generate a false alarm or incur a false negative error (undetected anomaly when it is present). This could be disastrous for the patients' health and may inflict a hard penalty upon the patterns provider. Therefore, injecting such malicious matrix should not happen in the context of SimilCare.

6. PERFORMANCE ANALYSIS

- **Computation cost.** In this section, we evaluate the computation performance of Π-CSP+ (Algorithm 2), which is the main core of the proposed Π-Knn protocol (Algorithm 3). This evaluation aims to analyze the effect of adding our privacy-preserving measurements through Π-CSP+ on the computational performance of the k-NN classifier. To do so, we consider a global context where a SimilCare HAR service denoted SC monitors a patient all day long. SC extracts (v) vectors of (n) features from the patient tracked signs and constructs the matrix M_B (Algorithm 2: instruction 4). Let SP denote a pattern service provider that holds (p) patterns of activities, from which it constructs the matrix M_A (Algorithm 2: instruction 2). Assume each pattern has (n) features and corresponds to the feature vectors extracted from the patient signs. Assume SP and SC run Π-CSP+ (Algorithm 2) by inputting M_A and M_B respectively. First, we assess the effect of HAR parameters that affect the performance of Π-CSP+ by construction, which are the number of features (n), the number of extracted vectors (v) and the number of patterns (p) held by the service provider. For this, we make three experiments denoted E_1, E_2 and E_3 and we vary n, v and p respectively in real values ranges chosen from literature works. Next, we evaluate the effect of the selected time window length on the performance of Π-CSP+. We select two representative window lengths to simulate the recognition of several types of activities and we perform E_4 and E_5 experiments. We describe the whole evaluation system in Fig. 5 and we provide each experiment's detail next in this section.

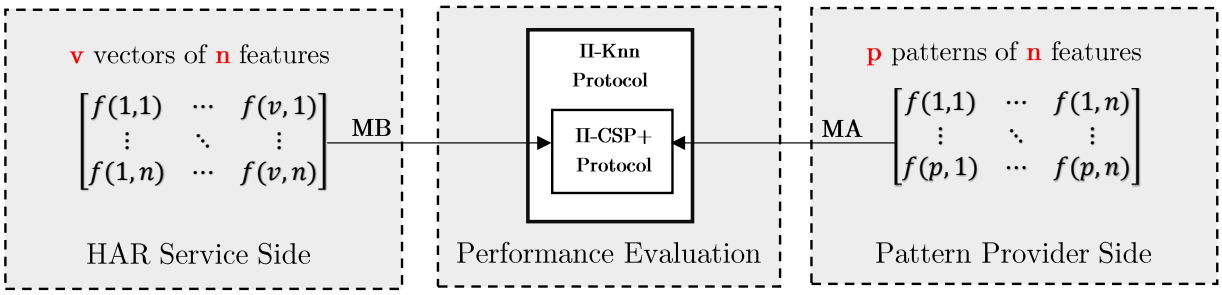


Figure 5: The performance evaluation system.

- **Communication cost.** Regarding the communications, one can observe that Π-CSP+ does not induce additional messages while integrating the noise within original data. In order to check this observation, let SP and SC denote the two parties executing Π-CSP+ and having respectively $(p \times n)$ and $(n \times v)$ vectors. Thus, according to Π-CSP+ (Algorithm 2), both SP and SC will communicate respectively M_{RA} and M_{RAB} having the same number of vectors ($p \times n$ and $n \times v$) as their original sets yet preserving their privacy.

- **Comparative evaluation.** Recall from section 2.3 that almost all existing privacy-preserving similarity evaluation protocols maintain a balance between data privacy and computational efficiency. Some protocols are built upon cryptographic schemes, such as Homomorphic encryption [25, 26, 27]. Such building blocks involve time-expensive operations that induce a high computational overhead and make them far less efficient in computational time. Other works present more efficient protocols than cryptographic ones with less privacy guarantees than our Π-CSP+ [24, 33].

  To the best of our knowledge, PCSC [4] is the most recent protocol for the private cosine similarity evaluation that provides a good privacy protection without involving cryptographic operations. Thus, we present a performance comparison between Π-CSP+ and PCSC protocol by performing the same experimental tests that we present next in this section.

  Moreover, to fairly compare the overhead incurred by the two protocols, we run again the same experimental tests using DCSC, the Direct Cosine Similarity Computation that does not involve any privacy-preserving measurement (equation (1), section 3.3). DCSC will serve as a running time reference for the two evaluated protocols.

6.1. Experimental environment and scenarios

We make experiments on the same set of vectors using a custom simulator built in Python and an Intel i5-2557M CPU running at 1.70 GHz and having a 4 GB of RAM. Thereafter, we describe the experimental scenarios.

Note 5. *In each experiment, we vary the evaluated parameter in a range of values to cover several literature works. For this raison, we use random generated observations instead of a real data set, which obviously has been got using fixed parameters.*

- **Experiment E_1: the effect of (n).** In this evaluation we test the effect of the number of features (n) involved within the vectors on the running time of our protocol (Π-CSP+), then, we make the same test on PCSC [4] and DCSC for comparison purpose. We fix the number of vectors (v) extracted by SC to 1 vector, and we fix the number of patterns held by SP to 500 patterns. Then, we vary the number of features (n) in the range $[10, 5000]$ features, to match several HAR real systems, whether they apply a dimensionality reduction phase or not. This scenario copes with Online-HAR [34], which provides a real-time recognition of the performed activity.

- **Experiment E_2: the effect of (v).** In this experiment, we test the effect of the number of vectors (v) extracted by SC and sent to SP on the running time of our protocol (Π-CSP+), then, we make the same test on PCSC [4] and DCSC for comparison purpose. For this, we fix the number of features to 40 features, which is the mean size used by several HAR systems [38, 39], and we fix the number of patterns held by SP to 500 patterns. Then, we vary v in the range $[10, 5000]$ vectors. This scenario matches Offline-HAR [34], where the classification is performed by a batch-process.

- **Experiment E_3: the effect of (p).** We make this experiment to test the effect of the number of patterns (p) held by SP on the running time of our protocol (Π-CSP+), then, we make the same test on PCSC [4] and DCSC for comparison purpose. We fix the number of features (n) to 40 features, and we fix the number of vectors (v) extracted by SC to 1 for online-HAR and 50 for offline-HAR (E_1 and E_2). Then, we vary p in the range [500,5000] patterns.

- **Experiments E_4 & E_5: the scalability assessment.** In this evaluation, we assess the scalability level of a pattern provider (SP) to handle online-HAR requests from several SimilCare clients (SC). For this, we consider SP receiving $N(w_i)$ similarity requests according to a Poisson process from different SimilCare clients at rate (λ) request(s) per time window (w_i): $N \sim P(\lambda)$. Assume the requests processing times (t_i) have an exponential distribution with rate (μ) requests per time window (w_i): $t_i \sim exp(\mu)$. We consider fixing the size of feature vectors to 40 features, which is the mean size of several HAR systems found in the literature. We consider the service provider having s processing servers with unlimited access, each of which holds 500 patterns, has a FIFO service discipline and operates all day long. Let M/M/s denote this system using Kendall's notation [40]. We make two experiments denoted E_4 and E_5 to simulate the case of one sever (s=1) and multi-server (s=20). We assess the usability rate (U) of the service provider and the waiting time (W) for each request to be served, besides other performance measures (the average number of clients in the system (L) and the average number of clients waiting in the queue (L_q)). Experiments E_4 and E_5 are performed according to different arrival rates ($\lambda \in \{10, 100, 500, 1000\}$). We use results of E_2 to determine the average processing rate (μ) of the evaluated protocol. Finally, we make comparison with PCSC protocol [4] and we sketch the results in Table 2, Fig. 8 and Fig. 9. Notice that in this simulation, we did not consider offline classification because there is no need to do it for different clients simultaneously.

  Selection of the window length (w). The computational complexity of any recognition system depends on the sampling window length used for feature extraction and classification. Several window lengths have been used in the literature ranging from w=1 second to w=45 seconds [41]. Selecting an appropriate length is a trade-off between the quality of the extracted features and the computational overhead. Generally, a long window length provides good extracted features [42]. However, the longer the window length is, the more the end-user should wait for the recognition result. Another trade-off involved in choosing a window length is the accuracy of the recognition result. That is, a short window may not provide sufficient information to recognize the performed activity. On the other hand, a long window may involve more than one activity within its range.

  Moreover, a main finding reported in several works is that the optimal window length depends on the activity to recognize [34, 42, 43]. It has been proved that a short window of $w \simeq$5 seconds allows the accurate recognition of posture and short duration activities, such as walking, sitting, ascending and descending stairs. Besides, a long window length on average of $w \simeq$30 seconds is adequate for long duration activities, such as making the bed, gardening and bathroom use [42].

  In this simulation, we use two different window lengths (w=5 seconds and w=30 seconds) to assess the scalability of our protocol for recognizing all types of activities.

6.2. Results and discussion

- **Results of E_1.** Through E_1 experiment, we evaluated the effect of varying the number of features (n) on the running time of the three similarity computation methods (Π-CSP+, PCSC and DCSC) in aim to evaluate online HAR. Results illustrated in Fig. 6a reveal the high efficiency level in running time of

Table 1: Performance measures of the simulated queueing models

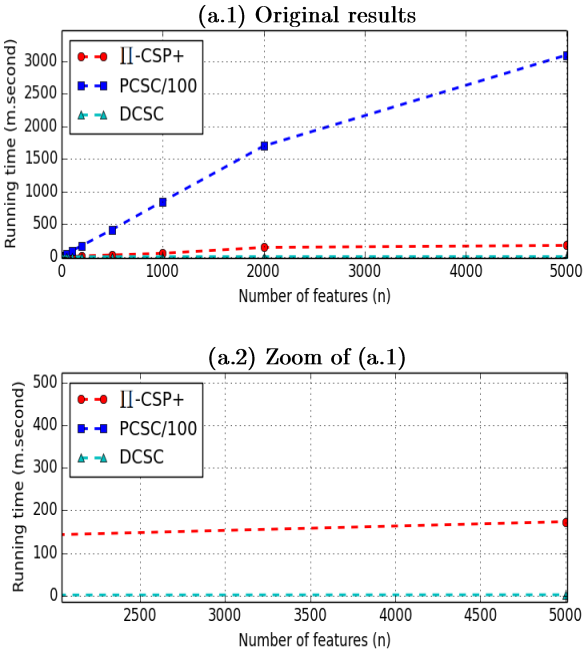
Performance Measures	**M/M/1 (E_4, s=1 server)**	**M/M/s (E_5, s=20 servers)**
(ρ) Intensity Traffic	$\frac{\lambda}{\mu}$	$\frac{\lambda}{s \times \mu}$
(U) Usability Rate	ρ	ρ
(P_0) Prob. System is idle	1 - ρ	$\left(1 + \frac{(s\rho)^s)}{s!(1-\rho))} + \sum_{n=1}^{s-1} \frac{(sn)^n)}{n!}\right)^{-1}$
(B) Prob. Queue Non-Empty	ρ^2	$\frac{(s\rho)^s}{s!(1-\rho)} P_0$
(L) Average Clients in the System	$\frac{\rho}{1-\rho}$	$s\rho + \rho\frac{B}{1-\rho}$
(L_q) Average Clients in the Queue	$\frac{\rho^2}{1-\rho}$	$\rho\frac{B}{1-\rho}$
(W) Average Waiting Time	$\frac{1}{\mu(1-\rho)}$	$\frac{1}{\mu}\left(1 + \frac{B}{s(1-\rho)}\right)$

Π-CSP+ that reaches 190 m.seconds for a very large number of features ($n = 5000$). Between $n = 30$ and $n = 50$, which is the average number of feature used by several real systems, Π-CSP+ had a slow increasing rate of around 37%. On the other hand, PCSC running time revealed a high increasing rate of 3600% between $n = 30$ and $n = 50$, besides a high overhead distance from the running time of DCSC computation reference (> 31000 m.seconds at $n = 1000$), which is highly greater than Π-CSP+ time distance from DCSC (< 50 m.seconds at n=1000). The efficient increasing rate in computation time of Π-CSP+ (37%) compared to PCSC (3600%) is due to the construction of Π-CSP+, where the random values added by the server to obfuscate its patterns are independent from the number of features involved in each pattern (Matrix MA, Algorithm 2). In contrast, PCSC protocol requires adding random values to each feature attribute.

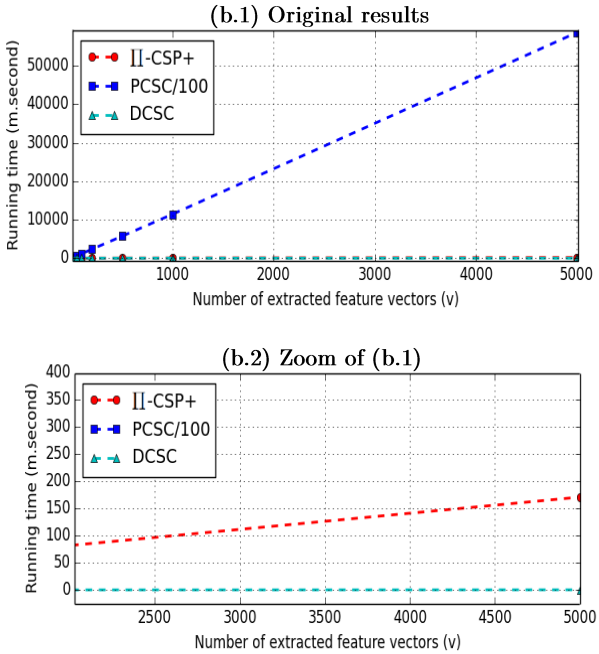
- **Results of E_2.** In E_2 experiment, we evaluated offline HAR by focusing on the effect of sending simultaneously v vectors on the running time of the previous three computation methods. Results shown in Fig. 6b reveal more clearly the efficiency in runing time of Π-CSP+. PCSC distance time from DCSC reference was increasing continuously (on average of 6000×100 m.seconds for v=500) with an increasing rate of around 4800% between v=500 and v=1000. On the other hand, Π-CSP+ presented a very efficient increasing rate of around 50% between v=500 and v=1000, while keeping a short stable distance on average of 30 m.seconds from DCSC running time (for $v < 2000$). The efficient increasing rate in computation time of Π-CSP+ (50%) compared to PCSC (4800%) is due to the construction of Π-CSP+, where the client side ensures the privacy of its vectors by its matrix parameters (section 5.4) and does not require to add any noise that is added on the server side. On the other hand, PCSC requires adding random values both on the server and the client side.

- **Results of E_3.** In this evaluation, we tested the effect of the number of patterns (p) on the running time of the previous similarity protocols. We avoided to plot the running time of DCSC as it had the same behaviour as in E_2. Results shown in Fig. 7a and Fig. 7b reveal a significant effect on the running time of Π-CSP+ compared to E_1 and E_2, which is due to the random matrix added by the server, and that relies on the number of patterns (p). Nevertheless, Π-CSP+ still provides an efficient response (< 200 m.seconds) compared to PCSC protocol, which was far less efficient with a response time that reached around 6000×100 m.seconds.

- **Results of E_4.** In E_4 experiment, we simulated a queueing model where a pattern provider (SP) serves multi SimilCare users for the HAR classification task. Assume SP having one processing server (s=1), we evaluated the queuing system using performance measures of M/M/1 model (Table 1)

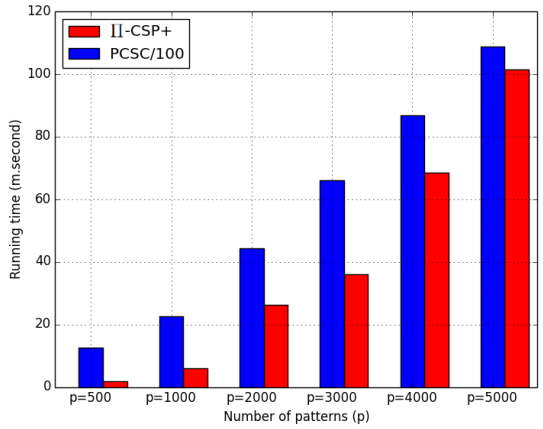


(a) E_1. Online HAR. Effect of the number of features (n) on running time ($v = 1, p = 500$)

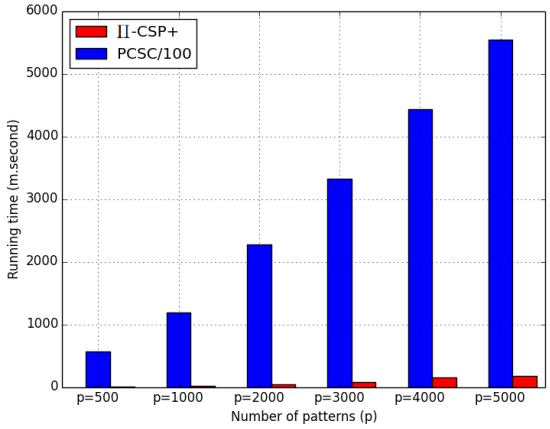


(b) E_2. Offline HAR. Effect of the number of vectors (v) on running time ($n = 40, p = 500$)

Figure 6: Evaluation of the effect of n and v on the HAR running time



(a) E_3. Effect of the number of patterns (p) on running time in online HAR ($v = 1, n = 40$)



(b) E_3. Effect of the number of patterns (p) on running time in offline HAR ($v = 50, n = 40$)

Figure 7: Evaluation of the effect of p on the HAR running time

according to a time window w=5 seconds. From evaluation results (Fig. 8, Table 2), we see that the server running PCSC protocol has not reached the steady state ($\rho > 1$) for all arrival rates ($\lambda \in \{10, 100, 500, 1000\}$), contrary to Π-CSP+, which was undergoing a slow intensity traffic ($\rho < 0.014$) due to its high processing rate ($\mu = 71400/s$). Π-CSP+ revealed a low usability rate ($U < 2\%$), which

Table 2: Evaluation results of the simulated queueing models in E_4 and E_5 experiments

Window Length (w)	**The Evaluated Protocol**	**Processing Rate** (μ)	**Arrival Rate** (λ/w_i)	**Intensity Traffic** (ρ)	**Usability Rate (U) %**	**Average Clients in the Syst.** (L) $\times$ $\mathbf{10^3}$	**Average Clients in the Queue** (L_q) $\times$ $\mathbf{10^6}$	**Average Waiting Time** (W) **ms**
5 (E_4)	Π-CSP+	71400	10	0.0001	0.01	0	0	0.01
			100	0.0014	0.14	1	2	0.01
			500	0.007	0.7	7	49	0.01
			1000	0.014	1.4	14	199	0.01
5 (E_4)	PCSC	4	10	>1	>100	∞	∞	∞
			100	>1	>100	∞	∞	∞
			500	>1	>100	∞	∞	∞
			1000	>1	>100	∞	∞	∞
30 (E_5)	Π-CSP+	400000	10	0.0000	0.00	0	0	0.002
			100	0.0000	0.00	0	0	0.002
			500	0.0000	0.00	1	0	0.002
			1000	0.0001	0.01	2	0	0.002
30 (E_5)	PCSC	26	10	0.0192	1.92	384	0	38.5
			100	0.1923	19.23	3846	0	38.5
			500	0.9615	96.15	39438	20207	78.9
			1000	>1	>100	∞	∞	∞

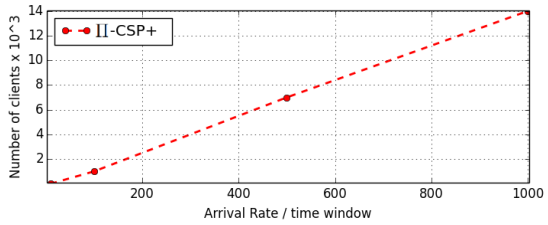
results in a very low probability of server overload. This claim may be confirmed by looking the average number of requests waiting in the queue ($L_q < 200 \times 10^{-6}$) that tends to 0 and induces a high efficient and stable response time ($W = 0.01$ m.seconds). Results of E_4 revealed the adequacy of Π-CSP+ to provide a recognition service each 5 seconds (w), which is the observation window adequate for posture activities. The recognition is provided for several SimilCare users (up to 1000), with only one server.

- **Results of E_5.** In E_5, we enlarged the observation window (w) to 30 seconds, which is adequate for household activities, and we assume the pattern provider (SP) having several processing servers (s=20). Then, we evaluated the queuing system using performance measures of M/M/s model (Table 1) according to the new time window (w=30 seconds). Evaluation results (Fig. 9, Table 2) show that the server running PCSC protocol could reach the steady state only for low arrival rates ($\lambda < 1000$ requests/30 seconds). After that, the server had been overloaded with a utilization rate of $U > 100\%$ and an infinite number of requests waiting in the queue ($L_q = \infty$). On the other hand, the server running Π-CSP+ becomes more efficient due to its high processing rate (μ=400000). Π-CSP+ had a very low utilization rate (U=0.01%) for a high arrival rate (λ=1000). Besides, it provided a very slow and stable response time ($W = 0.002$ m.seconds), which results in an empty waiting queue ($L_q = 0$). Results of E_5 have confirmed the adequacy of Π-CSP+ protocol to be implemented on servers that operates with several clients. Π-CSP+ handles similarity requests in an few time and scales efficiently for up to 1000 clients.

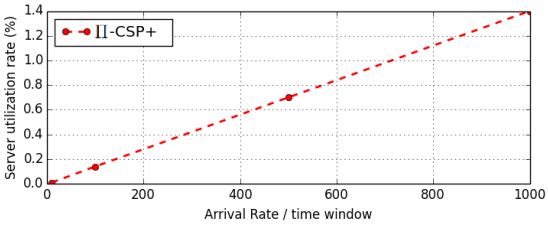
7. CONCLUSION

In this paper, we have proposed a secure k-NN classification protocol named (Π-Knn), designed for Human Activity Recognition (HAR). We have built this protocol on a novel efficient and privacy-preserving cosine similarity protocol named (Π-CSP+). As an application, we have integrated our proposed privacy-preserving HAR classifier in SimilCare, a novel medical monitoring framework, to support the medical decision by securely providing information about patients' activities. Through security analysis conducted with the standard real/ideal paradigm, we have proved the privacy protection of our proposed protocols that resist against malicious attacks. Across different experimental analysis performed with common used HAR parameters, our protocol (Π-CSP+) reached 37% overhead in Online-HAR and 50% overhead in Offline-HAR, which is high-efficient compared to other private similarity evaluation protocols from the literature. On the server side, Π-CSP+ provided a stable and efficient response time (W= 0.0x m.seconds) for both short and long time window lengths ($w \in \{5, 30\}$ seconds). All experimental results have confirmed the

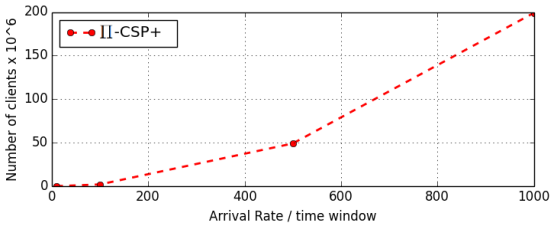
adequacy of our protocol for applications that require a real-time activity recognition service, as the medical monitoring.



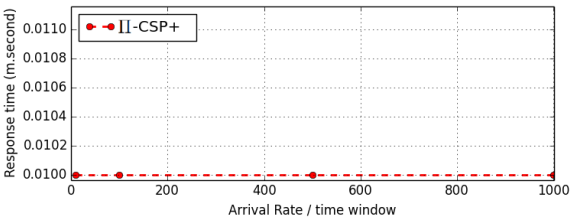
(a) Average Clients in the system (L)



(b) Usability Rate (U)

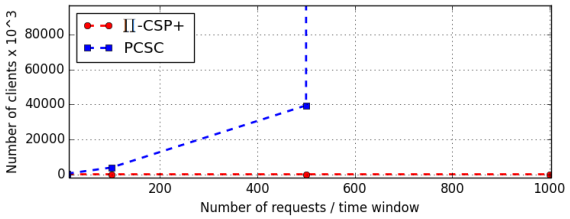


(c) Average Clients in the queue (Lq)

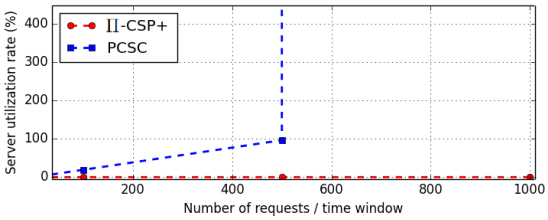


(d) Average Waiting time (W)

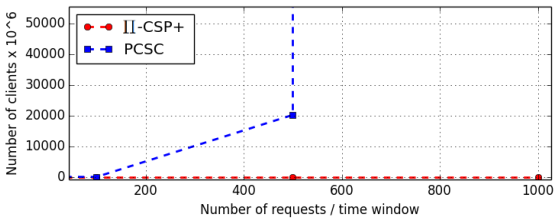
Figure 8: E_4. Performance results of a service-provider simulated with M/M/s model with s=1 server and a time window w=5 seconds (adequate for posture activities).



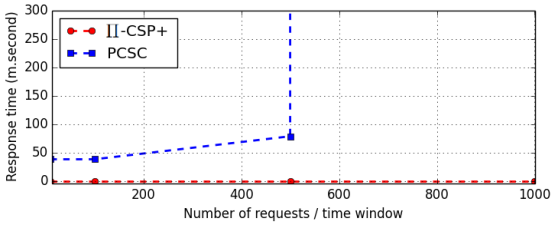
(a) Average Clients in the system (L)



(b) Usability Rate (U)



(c) Average Clients in the queue (Lq)



(d) Average Waiting time (W)

Figure 9: E_5. Performance results of service-provider simulated with M/M/s model with s=20 servers and a time window w=30 seconds (adequate for household activities).

REFERENCES

[1] A. Sachan, D. Roy, P. V. Arun, An Analysis of Privacy Preservation Techniques in Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 119–128, 2013.

[2] Z. Xu, X. Yi, Classification of privacy-preserving distributed data mining protocols, in: 2011 Sixth International Conference on Digital Information Management, pp. 337–342, 2011.

[3] Y. A. A. S. Aldeen, M. Salleh, M. A. Razzaque, A comprehensive review on privacy preserving data mining, SpringerPlus 4 (1) pp. 1, 2015.

[4] R. Lu, H. Zhu, X. Liu, J. K. Liu, J. Shao, Toward efficient and privacy-preserving computing in big data era, IEEE Network 28 (4) pp. 46–50, 2014.

[5] D. Malan, T. Fulford-Jones, M. Welsh, S. Moulton, Codeblue: An ad hoc sensor network infrastructure for emergency medical care, in: MobiSys 2004 Workshop on Applications of Mobile Embedded Systems (WAMES'04), Boston, MA, USA, 2004.

[6] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, Alarm-net: Wireless sensor networks for assisted-living and residential monitoring, Technical Report CS-2006-01; Department of Computer Science, University of Virginia: Charlottesville, VA, USA, 2006.

[7] R. Chakravorty, A programmable service architecture for mobile medical care, in: Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), Pisa, Italy, 2006.

[8] A. T. van Halteren, R. G. A. Bults, K. E. Wac, D. Konstantas, I. A. Widya, N. T. Dokovski, G. T. Koprinkov, V. M. Jones, R. Herzog, Mobile patient monitoring: The mobihealth system, The Journal on Information Technology in Healthcare 2 pp. 365–373, 2004.

[9] P. Kumar, H.-J. Lee, Security issues in healthcare applications using wireless medical sensor networks: A survey, Sensors 12 (1) pp. 55–91, 2011.

[10] M. A. Musen, B. Middleton, R. A. Greenes, Clinical Decision-Support Systems, Springer London, London, pp. 643–674, 2014.

[11] L. LL, Error in medicine, JAMA 272 (23) pp. 1851–1857, 1994.

[12] R. Canetti, Security and composition of multiparty cryptographic protocols, Journal of CRYPTOLOGY 13 (1) pp. 143–202, 2000.

[13] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Büla, P. Robert, Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly, IEEE Transactions on Biomedical Engineering 50 (6) pp. 711–723, 2003.

[14] J. C. Hou, Q. Wang, B. K. AlShebli, L. Ball, S. Birge, M. Caccamo, C.-F. Cheah, E. Gilbert, C. A. Gunter, E. Gunter, et al., Pas: A wireless-enabled, sensor-integrated personal assistance system for independent and assisted living, in: IEEE Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability,HCMDSS-MDPnP, pp. 64–75, 2007.

[15] S. Jiang, Y. Cao, S. Iyengar, P. Kuryloski, R. Jafari, Y. Xue, R. Bajcsy, S. Wicker, Carenet: An integrated wireless sensor networking environment for remote healthcare, in: Proceedings of the 3rd International Conference on Body Area Networks, BodyNets '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 9:1–9:3, 2008.

[16] S. L. Lau, I. König, K. David, B. Parandian, C. Carius-Düssel, M. Schultz, Supporting patient monitoring using activity recognition with a smartphone, in: 2010 7th International Symposium on Wireless Communication Systems, pp. 810–814, 2010.

[17] A. Evani, B. Sreenivasan, J. Sudesh, M. Prakash, J. Bapat, Activity recognition using wearable sensors for healthcare, in: the 7th International Conference on Sensor Technologies and Applications (SENSORCOMM 2013), pp. 173–177, 2013.

[18] D. De, P. Bharti, S. K. Das, S. Chellappan, Multimodal wearable sensing for fine-grained activity recognition in healthcare, IEEE Internet Computing 19 (5) pp. 26–35, 2015.

[19] L. Xiong, S. Chitti, L. Liu, Mining multiple private databases using a knn classifier, in: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC '07, ACM, New York, NY, USA, pp. 435–440, 2007.

[20] R. Mynavathi, V. Bhuvaneswari, T. Karthikeyan, C. Kavina, K nearest neighbor classifier over secured perturbed data, in: 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), pp. 1–4, 2016.

[21] Y. Qi, M. J. Atallah, Efficient privacy-preserving k-nearest neighbor search, in: 2008 The 28th International Conference on Distributed Computing Systems, pp. 311–319, 2008.

[22] W. K. Wong, D. W.-l. Cheung, B. Kao, N. Mamoulis, Secure knn computation on encrypted databases, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD '09, ACM, New York, NY, USA, pp. 139–152, 2009.

[23] E. Vani, S. Veena, D. J. Aravindar, Query processing using privacy preserving k-nn classification over encrypted data, in: 2016 International Conference on Information Communication and Embedded Systems (ICICES), pp. 1–5, 2016.

[24] J. Vaidya, C. Clifton, Privacy preserving association rule mining in vertically partitioned data, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 639–644, 2002.

[25] B. Goethals, S. Laur, H. Lipmaa, T. Mielikäinen, On Private Scalar Product Computation for Privacy-Preserving Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 104–120, 2005.

[26] H. Kikuchi, K. Nagai, W. Ogata, M. Nishigaki, Privacy-preserving similarity evaluation and application to remote biometrics authentication, in: Proceedings of the 5th International Conference on Modeling Decisions for Artificial Intelligence, MDAI '08 Sabadell, Springer-Verlag, Berlin, Heidelberg, pp. 3–14, 2008.

[27] D. Yang, B. Xu, B. Yang, J. Wang, Secure Cosine Similarity Computation with Malicious Adversaries, Springer New York, New York, NY, pp. 529–536, 2013.
[28] H. Huang, T. Gong, P. Chen, R. Malekian, T. Chen, Secure two-party distance computation protocol based on privacy homomorphism and scalar product in wireless sensor networks, Tsinghua Science and Technology 21 (4) pp. 385–396, 2016.
[29] Y. Zhu, Z. Wang, B. Hassan, Y. Zhang, J. Wang, C. Qian, Fast Secure Scalar Product Protocol with (almost) Optimal Efficiency, Springer International Publishing, Cham, pp. 234–242, 2016.
[30] W. Du, M. Atallah, Privacy-preserving cooperative statistical analysis, in: Proceedings of the 17th Annual Computer Security Applications Conference, ACSAC '01, IEEE Computer Society, Washington, DC, USA, pp. 102, 2001.
[31] W. Jiang, M. Murugesan, C. Clifton, L. Si, Similar document detection with limited information disclosure, in: 2008 IEEE 24th International Conference on Data Engineering, pp. 735–743, 2008.
[32] M. Murugesan, W. Jiang, C. Clifton, L. Si, J. Vaidya, Efficient privacy-preserving similar document detection, The VLDB Journal 19 (4) 457–475, 2010.
[33] I. Leontiadis, M. Önen, R. Molva, M. J. Chorley, G. B. Colombo, Privacy preserving similarity detection for data analysis, in: 2013 International Conference on Cloud and Green Computing, pp. 547–552, 2013.
[34] O. D. Lara, M. A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Communications Surveys & Tutorials 15 (3) pp. 1192–1209, 2013.
[35] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowledge and Information Systems 14 (1) pp. 1–37, 2008.
[36] A. M. Qamar, E. Gaussier, J.-P. Chevallet, J. H. Lim, Similarity learning for nearest neighbor classification, in: Eighth IEEE International Conference on Data Mining ICDM'08, pp. 983–988, 2008.
[37] Y. Lindell, B. Pinkas, Secure multiparty computation for privacy-preserving data mining, Journal of Privacy and Confidentiality 1 (1) pp. 59–98, 2009.
[38] K. Altun, B. Barshan, Human Activity Recognition Using Inertial/Magnetic Sensor Units, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 38–51, 2010.
[39] J. Cheng, O. Amft, P. Lukowicz, Active Capacitive Sensing: Exploring a New Wearable Sensing Modality for Activity Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 319–336, 2010.
[40] E. Gelenbe, G. Pujolle, J. Nelson, Introduction to queueing networks, Vol. 2, John Wiley & Sons, Inc., 1987.
[41] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, Sensors 14 (4) pp. 6474–6499, 2014.
[42] E. Munguia Tapia, Using machine learning for real-time activity recognition and estimation of energy expenditure, Ph.D. thesis, Massachusetts Institute of Technology, 2008.
[43] T. Huynh, B. Schiele, Analyzing features for activity recognition, in: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies, ACM, pp. 159–163, 2005.