A Privacy-Preserving-based Data Collection and Analysis Framework for IoMT Applications

Muhammad Usman, *Member, IEEE,* Mian Ahmad Jan, *Member, IEEE,* Xiangjian He, *Senior Member, IEEE* and Jinjun Chen, *Senior Member, IEEE*

***Abstract*—The concept of Internet of Multimedia Things (IoMT) is becoming popular nowadays, and can be used in various smart city applications, such as traffic management, e-health, and surveillance. In IoMT applications, multimedia data is generated from devices with sensing capabilities, e.g., Multimedia Sensor Nodes (MSNs). These devices come with limited computational and storage resources, and cannot hold captured multimedia data for a long time if the connection between a base station and cloud server is down. In this situation, mobile sinks can be utilized to collect data from MSNs and upload to the cloud server. However, such a data collection raise privacy issues, such as revealing identities and location information of MSNs to mobile sinks. Therefore, there is a need to preserve the privacy of MSNs during mobile data collection. In this paper, we propose an efficient privacy-preserving-based data collection and analysis framework for IoMT applications. The proposed framework distributes an underlying Wireless Multimedia Sensor Networks (WMSNs) into multiple clusters. Each cluster is represented by a cluster head. The cluster heads are responsible to protect the privacy of member nodes through data and location aggregation. Later, aggregated multimedia data is analyzed at a cloud server using a counter-propagation artificial neural network to extract meaningful information through segmentation. Experimental results show that our proposed framework outperforms other state-of-the-art techniques, and can be used to collect multimedia data in various IoMT applications.**

***Index Terms*—IoMT, MSNs, privacy, WMSN, clusters, counter-propagation artificial neural network.**

I. Introduction

RECENT developments in the electronic industry have enabled sensing devices to capture high-resolution multimedia, and transformed the concept of Internet of Things (IoT) to Internet of Multimedia Things (IoMT) [1]. In an IoMT framework, devices are able to capture and sense multimedia and non-multimedia data, respectively. A most common example of these devices is Multimedia Sensor Nodes (MSNs). Just like simple sensor nodes, these nodes also form a network known as Wireless Multimedia Sensor Network (WMSN) [2]. Just like traditional wireless sensor networks, captured multimedia data in a WMSN are forwarded to a nearby Base Station (BS) to perform computationally expensive tasks and offload processed data to a cloud server. The BS is connected to the cloud server through a high-speed Internet connection. However, it is possible that the BS is unable to upload processed multimedia data to the cloud server due to technical problems in underlying telecommunication network. In this particular situation, mobile sinks can be utilized to collect data from nominated MSNs, known as Cluster Heads (CHs), and forward to the cloud server. Such a data collection is also referred as mobile crowdsensing [3], where a mobile device, such as a smartphone and tablet, can sense, compute, and share data with other nearby devices for a common interest. However, involvement of mobile sinks for data collection in sensitive IoMT applications may raise issues like protecting the privacy of original data source (i.e., MSNs).

In a mobile data collection scenario, credibility of mobile sinks becomes a challenging task when they are anonymous. Sharing data with mobile sinks requires a proper authentication/registration. Without a proper authentication/registration, an intruder can bring down an entire network by stealing identities of member nodes and manipulating data exchange between member nodes. Even if mobile sinks are authenticated, still there are chances that privacy of devices of an underlying network may be compromized using malicious applications running on mobile sinks [4]. IDs and location information of corresponding MSNs can easily be determined by analyzing the shared data. Furthermore, MSNs may be a part of sensitive applications, such as e-health, surveillance, and transport management. If compromised, an attacker can not only manipulate the forwarded data to generate misleading results, but also can gain access and control the underlying network. Therefore, protecting the privacy of MSNs becomes an important and serious security concern in IoMT applications.

In recent years, many machine learning techniques are used in networking domain to offer various services, such as providing security and privacy. Popular algorithms, such as k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), and Artificial Neural Network (ANN), were used to develop intrusion detection systems [5]. Similarly, many privacy-preserving techniques based on machine learning algorithms were proposed to protect location information of participating devices [6]–[9]. In these techniques, naive approaches are used to hide sensitive information of participating devices and release non-sensitive data. However, an action of protecting the sensitive information can still infer hidden information. It was shown that human activities and behaviors in specific situations can easily be modeled through Markov chains [10], [11]. As a result, an adversary can easily find a temporal correlation between actions and situations, and make afore-

Muhammad Usman and Jinjun Chen are with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Australia. (E-mail: musman@swin.edu.au, j.chen@swin.edu.au

Xiangjian He is with the Global Big Data Technologies Center (GBDTC), School of Electrical and Data Engineering, University of Technology Sydney, Australia. (E-mail: xiangjian.he@uts.edu.au.

Mian Ahmad Jan is with the Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan (E-mail: mianjan@awkum.edu.pk)

mentioned privacy-preserving approaches fail. Furthermore, applying complex machine learning algorithms on resource-constrained devices (e.g., MSNs and mobile devices) to offer privacy and security solutions is not a good idea at all. This problem can easily be solved by applying a simple privacy-preserving technique in IoMT applications to protect the privacy of participating devices during a data collection, and utilizing complex machine learning algorithms on cloud platforms for further analysis.

In this paper, we propose a privacy-preserving-based data collection and analysis framework for IoMT applications. The proposed framework operates in two phases. In the first phase, an underlying WMSN is divided into multiple clusters and CHs are selected to collect data from member nodes. Secondly, mobile sinks are registered with the BS. Once registered, the mobile sinks are allowed to collect data from CHs. These CHs play the role of intermediate devices between mobile sinks and member nodes. Instead of applying complex privacy-preserving algorithms, data and location information are aggregated at CHs before sharing with mobile sinks. This scenario can be considered as a special case of local differential privacy, in which original data and location information are manipulated by introducing errors before uploading to the cloud servers. These errors are called noise. In our proposed framework, aggregated data and location information give an impression to mobile sinks that the data is coming from a single entity (i.e., CHs), not from a section of a network. As a result, the mobile sinks may assume that a CH can be the original source or just a relay node. Furthermore, if an adversary tries to analyze the aggregated data, it is not easy to determine the original data and location information of original sources, i.e., MSNs. In the second phase, mobile sinks forward collected multimedia data to cloud server. Once received, a Counter-Propagation Artificial Neural Network (CP-ANN) is applied to segment foreground and background regions. To the best of our knowledge, there is no existing framework for an IoMT paradigm to preserve the privacy of member nodes using data and location aggregation and analyze the aggregated data on cloud servers using a machine learning technique. Major contributions of our proposed framework are as follows.

- A lightweight handshaking mechanism is applied to divide an underlying WMSN into multiple clusters. These clusters are represented by CHs. In each round of simulation, new CHs are selected based on their current energy levels. These CHs are responsible for many activities, such as node authentication, data collection and aggregation, and sharing the aggregated data with mobile sinks. A similar handshaking mechanism is also used to register mobile sinks with BS.
- A lightweight aggregation technique is applied by CHs on the received multimedia data and location information. Such an aggregation helps in protecting the sensitive information of member nodes, such as angular position, location information, and IDs. If the sensitive information got leaked, an adversary can easily identify the exact location of data source, enter the network, and cause serious damages.
- A CP-ANN is used on cloud server to process the aggregated data, and segment foreground and background regions to extract meaningful information, e.g., tracking moving objects and identifying malicious activities. The aggregated data is compressed on mobile sinks using a video coding standard before forwarding to the cloud server. Experimental results show that our proposed framework is still able to extract the required information from the received compressed data.

The rest of this paper is structured as follows. Section II provides a literature review on efforts made in recent years. The proposed system is explained in Section III. Experimental setup and simulation results are discussed in Section IV. Finally, the paper is concluded in Section V.

II. Literature Review

We distribute this section into two subsections. Both subsections discuss various privacy-preserving techniques. In the first subsection, we provide an overview of privacy-preserving techniques based on random methodologies. In the second subsection, the focus is on privacy-preserving techniques based on machine learning algorithms.

A. Privacy-Preserving using Random Methodologies

In recent years, concerns about security and privacy of sensing devices have stimulated interests in both academic and research industries. Most sensing devices usually operate in open wireless environments, and may become vulnerable to various security and privacy threats. In [12], a survey on security and privacy-related issues in vehicular ad-hoc networks was presented. In this survey, various methods addressing security and privacy challenges in vehicular devices are reviewed and explained for detection and revoking of malicious nodes. In [13], a survey on existing authentication and privacy-preserving schemes to secure 4G and 5G cellular networks was presented. In this survey, various schemes are discussed and analyzed from a perspective of four different types of attacks, i.e., privacy, integrity, availability, and authentication attacks. Surveys presented in [12], [13] highlight various security and privacy challenges for mobile devices which can be used to collect data from various sensitive IoMT applications.

In [14], a fusion of IoT, big data, and cloud storage was presented to preserve the privacy of sensory data collected from e-health systems. In this fusion, IoT group keys are used to authenticate medical nodes and encrypt messages in a batch processing style to minimize the computational time. In [15], an advanced framework for opportunistic routing schemes in delay-tolerant networks was proposed. In this framework, the main focus is to protect the confidentiality of nodes and perform anonymous authentication using a pairwise communication. In [16], a privacy-preserving sensory data collection scheme on an Internet of Vehicles was proposed. In this scheme, location privacy is preserved through modified Pailier cryptosystem during data aggregation, and a proxy re-encryption technique is used to preserve the privacy at network edge during data acquisition. Approaches presented in [14]–[16] are based on cryptography-based solutions and may not be suitable for real-time IoMT applications.

In [17], two practical schemes were proposed to protect the privacy of trust evidence providers using an additive homomorphic encryption. These schemes are evaluated against various internal attacks and proved to be efficient for various big data processing applications. In [18], an efficient privacy-preserving compressive data gathering scheme was proposed for a WSN. In this scheme, homomorphic encryption functions are exploited to trace network traffic and preserve the privacy of nodes by making message content confidential. Techniques presented in [17], [18] are efficient to stand against internal network attacks, but are not suitable for data collection using mobile sinks in IoMT applications.

B. Privacy-Preserving using Machine Learning Methodologies

In [19], a machine learning classification was performed using a fully homomorphic encryption. In this scheme, a hyperplane decision-based classification is combined with a Naive Bayes classification using additive and multiplicative homomorphic techniques without leaking user privacy in an outsourcing scenario. In [20], a differential Naive Bayes learning scheme with multiple data sources was proposed to prevent disclosure of sensitive information of data owners. In this scheme, a trainer is enabled to train a Naive Bayes classifier over provided data sets from different data owners to achieve the differential privacy for all data owners. In [21], a Naive Bayes classifier was used to preserve privacy against a substitution-then-comparison attack. In this approach, a double-blinding technique is used to combine an additively homomorphic encryption with an oblivious transfer to hide user privacy at low computational costs. Approaches presented in [19]–[21] are based on a combination of machine learning techniques and data encryption, and encryption of real-time multimedia data in IoMT applications cannot be considered an ideal solution to preserve the privacy of original data sources.

In [22], a client-server data classification protocol based on an SVM was proposed to preserve the privacy during a data classification at cloud side. In this protocol, properties of Pailer homomorphic encryption are exploited to ensure the privacy in multi-class problems, and securely obtain the sign of Pailer encrypted numbers. In [23], distributed machine learning algorithms based on alternating direction method of multipliers were proposed to preserve the privacy of a network. In these algorithms, dual and primal variable perturbations are used to provide dynamic differential privacy under mild conditions of convexity and differentiability of loss function and regularizer. In [24], a privacy-preserving machine learning based collaborative intrusion detection system for vehicular ad-hoc networks was proposed. In this system, an alternating direction method of multipliers and dual-variable perturbation are used to train a classifier to detect intruders and provide dynamic differential privacy, respectively. In [25], a hybrid scheme was combined with deep learning to preserve the privacy of data stored on cloud servers. In this combination, a multi-key homomorphic encryption is combined with double decryption mechanisms, fully homomorphic encryptions, and deep learning schemes to preserve the privacy of encrypted data. In [26], a privacy-preserving sparse representation classification technique was proposed for cloud-enabled mobile applications. In this technique, the privacy of data contributors and application users is protected in the presence of an untrusted cloud server against various types of attacks, such as content privacy attack, source privacy attack, and label privacy attack. Schemes presented in [22]–[26] are able to stand against various types of privacy attacks, however, these schemes are not suitable for real-time IoMT applications.

III. Privacy-Preserving-based Data Collection and Analysis

In this section, we explain our proposed Privacy-Preserving-based Data Collection and Analysis framework (P2DCA) for IoMT applications. A block diagram of our proposed framework is shown in **Fig. 1**. Our proposed framework is not only able to protect location privacy of member nodes using data and location information aggregation, but also can analyze multimedia data using a CP-ANN to extract meaningful information on the cloud server. The aggregation is required to ensure that no one can locate the original data source in a WMSN. Our proposed framework is divided into two phases, i.e., location privacy protection, and data analysis. In the following subsections, these phases are explained in detail.

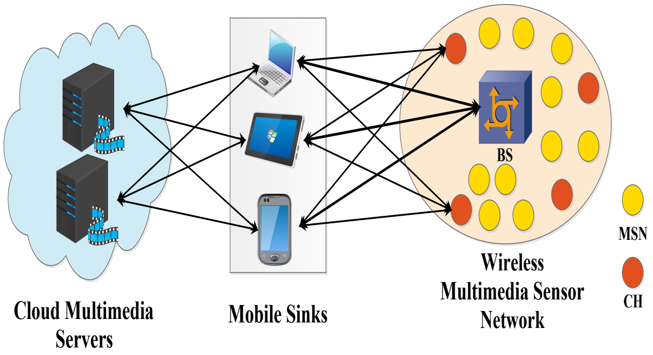

Fig. 1: Data Collection and Analysis Framework

A. Location Privacy Protection

In our proposed framework, privacy of member nodes is protected through the k-anonymity rule. In [27], the concept of k-anonymity was proposed to protect individual identities of subjects participating in a data collection. According to the definition, reports from k-sources should be combined together before forwarding. Based on this definition, a WMSN is divided into multiple clusters. Each cluster consists of at least k member nodes, out of which one is selected as a CH. The selected CHs are responsible to collect data from member nodes and forward to BS for further processing. The BS is responsible to perform heavy computational tasks and forward the processed data to cloud server. In our proposed framework, we assume that the connection between BS and cloud server is down. As a result, the BS is unable to forward the processed data to cloud server. In this situation, mobile sinks can collect data from CHs and forward to cloud server. There are m mobile sinks in nearby locations of member nodes, where $m \in \{1, 2, \cdots, M\}$. However, sharing information like locations of member nodes and architecture of an underlying WMSN with mobile sinks is not safe at all. Therefore, a trusted third party is required to protect the privacy of an underlying WMSN. In our proposed framework, there are two trusted third parties, i.e., BS and CHs. Operation of this phase is summarized in the following subsections.

1) Information Sharing: In each round of simulation, an MSN (i.e., α) in an underlying WMSN forwards its private information, such as ID (i.e., i where $i \in \{1, 2, \cdots, I\}$), location coordinates (i.e., (x_i, y_i)), and current energy level (i.e., e_i) to BS. The MSNs are randomly deployed in the underlying network. Their private information is encapsulated in a request packet. Upon reception, the BS extracts all embedded information, stores in its database, and computes the average energy threshold to decide which nodes are going to be the CHs in current round of simulation. The average energy threshold (i.e., E) is computed using the following equation.

$$E = \sum_{i=1}^{I} \frac{e_i}{I}. \tag{1}$$

After computing E, it is compared with the energies of all member nodes. The MSNs having energy equal or greater than the average energy threshold are eligible to be selected as CHs. There may be a situation in a specific simulation round where multiple MSNs have the same energy levels. In such a situation, selection is done based on very minor differences in the shared energy levels, and can go up to four decimal places. It is also possible that some eligible MSNs have exactly the same energy levels. In this specific scenario, the selection of CHs is based on a factor that an α_i is not selected as a CH in past four rounds.

2) Cluster Formation: After finalizing CHs (i.e., βs) in a particular simulation round, BS broadcasts a message containing IDs (i.e., j, where $j \in \{1, 2, \cdots, J\}$ and $J \subseteq I$) and location coordinates (i.e., (x_j, y_j)) of selected βs in the entire WMSN. Upon reception, following operations are performed by a CH, 1) retrieves appended IDs and location coordinates of selected CHs and stores in its buffer, 2) sends an acknowledgment message back to BS to confirm the successful receiving of information, and 3) advertises itself to its neighbors by sharing its current energy level and load capacity. The main motive behind storing information of neighboring CHs is to coordinate with them for multiple reasons, such as load sharing, hand-overing of nodes, etc. It is possible that an α_i may receive invitations from multiple βs. In this situation, an α_i associates itself with a β_j having the shortest distance, maximum energy level, and load capacity after verifying its ID from the received list forwarded by BS. The shortest distance (i.e., d) measurement is based on Euclidean distance, and estimated by the following equation.

$$d = \sqrt{(x - x_i)^2 - (y - y_i)^2}. \tag{2}$$

After estimating d, an α_i sends a joining request to the targeted β_j. It is possible that there are multiple βs satisfying the shortest distance criteria. In this situation, joining request is sent to all targeted βs. After approval, an α_i can associate itself with only one β at a time, and becomes a part of a cluster. This joining process requires mutual authentication to prove that both an α_i and β_j are trusted entities. This mutual authentication is based on our work proposed in [28]–[30].

3) Mobile Sink Registration: In a cluster-based data communication, CHs play the role of a trusted third party between member nodes and BS. They are responsible to collect data from member nodes and forward to BS for further processing and uploading to cloud server. In our proposed framework, we are assuming that the Internet connection between BS and cloud server is temporarily down, and BS is unable to upload processed data to cloud server. In this situation, mobile sinks can be utilized to collect data from CHs, and upload to cloud server. However, it is important to verify identities of mobile sinks before sharing the data. This verification is based on a two-step process, i.e., registration with BS and authentication with CHs.

As shown in **Fig. 2**, registration with BS is a four-step process. Here a mobile sink is represented by an Ω. In the first step, an Ω_m generates a session key (i.e., μ_m) and two random integers (i.e., a and b, where $a \in Z$ and $b \in Z$). An encrypted registration request (i.e., R_m) is generated by the following equation.

$$R_m = AES\{m, (H_2(a \oplus \mu_m))\}, \tag{3}$$

where H_2 is a mapping function, used to perform one-way secure hashing with a value range in $\{0, 1\}$ [31], and $\oplus$ is an XoR operator.

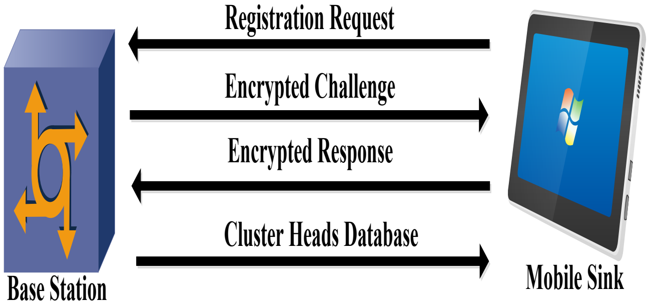

Fig. 2: Mobile Sink Registration

Upon reception, BS decrypts the request to extract embedded values. Based on forwarded information, BS generates an encrypted challenge (i.e., ϑ) by using **Eq. 4d** and forwards it to the requesting Ω_m.

$$\Delta_m = c.H_1(m \mid\mid H_2(a \oplus \mu_m)), \tag{4a}$$

$$\delta_m = H_2(H_2(m \mid\mid d)), \tag{4b}$$

$$\Psi_m = H_2(m \,||\, a \,||\, d), \tag{4c}$$

$$\vartheta = AES\{\Delta_m, \delta_m, \Psi_m\}, \tag{4d}$$

where, Δ_m is a registration certificate, H_1 is a hashing function used to perform a map to point hashing operation with a value range in $\{0, 1\}$ [31], δ_m is a time-stamp, Ψ_m is an authentication value, and c and d are random integers where $c \in Z$ and $d \in Z$.

The registration certificate is used for future communication, time-stamp represents the total amount of time a mobile sink can spend in the underlying WMSN, and authentication value is used to create a response. Upon receiving, an Ω_m decrypts ϑ to extract embedded values to create a response (i.e., $\overline{R_m}$) by using the following equation.

$$O_m = H_2(\Psi_m \,||\, H_2(a \oplus \mu_m) \,||\, b), \tag{5a}$$

$$\overline{R_m} = AES\{O_m, b\}. \tag{5b}$$

Upon receiving $\overline{R_m}$, BS verifies the identity through **Eq. 5a**. After verification, following operations are performed by BS, 1) sharing of Δ_m and δ_m with all βs, and 2) forwarding information of selected βs, such as js and (x_j, y_j), with registered Ω_m.

4) Coordinates Aggregation: In real-world scenarios, MSNs within a cluster usually focus on the same scene at different angles. As a result, the same information is captured and transmitted in bulk volumes and consumes computational resources and network bandwidth. Applying aggregation on such huge data may produce different results as compared to original data, however, it cannot be considered as a great loss. Furthermore, it is hard to estimate from aggregated data the exact angle at which the data are captured, and as a result, the actual source of data cannot be located and the privacy of individual sources is preserved. Furthermore, data aggregation helps in efficiently utilizing the available storage space at CHs and the bandwidth during data transmission.

In our proposed framework, CHs apply similarity-based data aggregation to minimize data redundancy and preserve the privacy of member nodes. The concept of data redundancy is taken from video coding. A video is a combination of multiple frames. In multimedia data transmission, videos are usually transmitted as a group of frames. Before transmission, complex video coding algorithms are applied to reduce size of a video by throwing away redundant information in video frames. In our proposed framework, a video from one of the MSNs within a cluster is used as a standard data once in a while. Videos from the remaining MSNs (i.e., repeated videos) within the same cluster are compared with the standard one as shown in **Fig. 3**. To find the similarity, the comparison is performed on a frame by frame basis. Each video frame is split into multiple blocks of equal size (i.e., 64 × 64). The blocks in a current video frame from a repeated video are compared against the blocks of a frame from the standard video. This comparison is performed using histogram normalization. If normalized histograms of blocks from current and standard video frames are represented by H_c and H_s, respectively, then the similarity value S can be estimated using the following equation.

$$S = H_s \times \log_2\left(\frac{H_s}{H_c}\right). \tag{6}$$

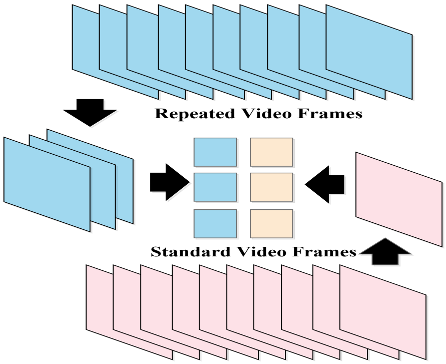

Fig. 3: Similarity-based Data Aggregation

The estimated S is compared against a predefined threshold $\overline{S}$ where $\overline{S} \in (S_{min}, S_{max})$. If $S < \overline{S}$, then the block is considered dissimilar otherwise considered similar. If 25% blocks are found dissimilar, then it is assumed that the current video frame contains important information and needs to be transmitted.

Location information in a cluster is also aggregated to preserve the privacy of member nodes, and as a result, the aggregated data correspond to aggregated locations. Such an aggregation of data and location help in preserving the privacy of member nodes based on a k-anonymity rule. Due to this aggregation, the destination (i.e., a mobile sink in our case) assumes that the data is coming from a single entity, not from k nodes. CHs aggregate location information by computing a mean value of location coordinates of member nodes. Later, the aggregated multimedia data is associated with aggregated location information and shared with a nearby mobile sink.

Although aggregation of data and location information help in preserving the privacy of member nodes, it also causes an information loss. To ensure a minimum information loss during a data aggregation, we assume that all member nodes in a cluster are located close to each other. This assumption can be verified for each member node using the following equations.

$$\theta = \sum_{i=1}^{k} (x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2, \tag{7}$$

$$\begin{aligned}\theta &= \sum_{i=1}^{k} \left(x_i - \frac{\sum_{g=1}^{k} x_g}{k}\right)^2 + \left(y_i - \frac{\sum_{g=1}^{k} y_g}{k}\right)^2 \\ &= \sum_{i=1}^{k} {x_i}^2 - \frac{(\sum_{g=1}^{k} x_g)^2}{k} + \sum_{i=1}^{k} {y_i}^2 - \frac{(\sum_{g=1}^{k} y_g)^2}{k},\end{aligned} \tag{8}$$

$$\Theta = \sum_{f=1}^{F} \theta_f, \tag{9}$$

where k represents total number of nodes in a cluster where $k \leq I$, (x_i, y_i) are coordinates of an α_i, $(\hat{x_i}, \hat{y_i})$ is the mean value of location coordinates of a cluster, θ is information loss per cluster, and Θ is total information loss of an entire network.

In **Eq. 7-9**, small values of θ and Θ mean less information loss when aggregation is applied to data. In traditional privacy-preserving frameworks based on local differential privacy, some noise is usually added to data to protect the privacy of original data sources. However, addition of noise may increase size of data, especially in the case of multimedia data. Therefore, it cannot be considered an ideal solution to preserve the privacy in scenarios containing multimedia data, low energy nodes, and limited bandwidth. In our proposed framework, we apply aggregation to original data to preserve the privacy of member nodes in the underlying network. This aggregation can be considered as noise, as it modifies the original data. Due to aggregation of data and location information, a compromised mobile sink cannot determine the actual data and location coordinates of member nodes.

B. Data Analysis

After receiving data from CHs, mobile sinks upload multimedia data to cloud server. However, before uploading, multimedia data need to be encoded to meet the available bandwidth requirement between mobile sinks and cloud server. For encoding, we apply a scalable video coding technique to generate video bitstreams with variable bitrates. Aggregated videos encoded in a scalable style make it challenging for data analysis applications on cloud servers to perform segmentation and detect moving objects. To efficiently detect moving objects from scalable coded videos, we use a CP-ANN on cloud server. This CP-ANN consists of two modules, i.e., Background Generation Module (BGM) for training and Object Extraction Module (OEM) for extracting objects. In general, a CP-ANN is a three-layer architecture where the first layer is an input layer, the second layer is a Kohonen layer, and the third layer is a Grossberg layer [32]. In this architecture, neurons from one layer are connected to neurons of next layer in sequence. Due to an unsupervised learning nature, each neuron in the Kohonen layer characterizes input patterns based on a winner-take-all rule. A winning neuron is selected based on longest distance between input and Kohonen layers. For each category, the Grossberg layer produces an output. Based on this working style, a BGM can predict properties of incoming video bitstreams by exploiting various properties of pixels in each video frame. In the following subsections, we explain modules of CP-ANN in detail.

1) Background Generation Module: A video frame is usually represented by one luminance (i.e., Y) and two chroma (i.e., C_b and C_r) components. If a pixel from an incoming video frame is represented by $p(x,y)$ where (x,y) represents coordinates of a pixel in a video frame and p_Y, p_{C_b} and p_{C_r} represent the luminance and chroma components as the input patterns in input layer, then the distance (i.e., Ψ) for a neuron between the input and Kohonen layers can be computed by the following equation.

$$\Psi_n = \frac{\sqrt{(p_c - v)^2}}{\sqrt{\sum(p_c)^2}} \qquad 1 \leq n \leq N, \tag{10}$$

where p_c represents p's component value in YC_bC_r color space, n and v represent ID and current weight of corresponding neuron in the Kohonen layer.

A winner neuron is selected based on a maximum distance as shown in the following equation.

$$\Psi_{max} = \max_{n=1,2,\cdots,N} \Psi_n. \tag{11}$$

To declare a winner neuron, an empirical tolerance (i.e., T) is used for comparison as shown in the following equation.

$$n = \begin{cases} winner, & if \quad \Psi_{max} < T \\ loser, & otherwise \end{cases}. \tag{12}$$

Once a winner neuron is selected, weights around its position in the Kohonen layer are adjusted using the following equation.

$$\hat{v} = v + l(p_c - v), \tag{13}$$

where $\hat{v}$ represents newly adjusted weight and l is learning rate set to 0.01 in our proposed framework.

Neurons between the Kohonen and Grossberg layers are connected on a one-on-one basis, therefore, weight (i.e., w) between these two layers is set to 1 to maintain the characteristics of each neuron. This entire unsupervised strategy helps us to determine properties of each video bitstream where the properties are characterized into neurons of the Kohonen layer. Such a characterization in this module helps in identifying moving objects and background segmentation in incoming video bitstream encoded in a scalable style.

2) Object Extraction Module: After determining a relationship between neurons in the input layer and Kohonen layers and deciding a winner neuron in the Kohonen layer, next step is to compute output for the Grossberg layer. If a winner neuron exists in the Kohonen layer, then output of the Grossberg layer is set to 1. If no winner neuron exists, then a similarity between a $p(x,y)$ and n is determined through a Gaussian function. If a similarity exists, then there is a high probability that a $p(x,y)$ belongs to background scene in a video frame, and is treated as a background pixel (i.e., $b(x,y)$). This Gaussian function based similarity estimation is used to estimate output of the Grossberg layer and can be computed by the following equation.

$$b(x,y) = \begin{cases} 1, & if \quad \Psi_{min} < T \\ w.e^{\frac{-(\sum_{n=1}^{N} p_c - v)}{T^2}}, & if \quad otherwise \end{cases}. \tag{14}$$

Unlike **Eq. 12**, empirical tolerance in **Eq. 14** is based on a minimum value of Ψ. A smaller value of T in **Eq. 12** generates new neurons in the Kohonen layer. On the other hand in **Eq. 14**, it generates a gradual Gaussian curve in the Grossberg layer. To avoid misjudgment of background pixels, we set a lower value (i.e., $T = 0.12$) in our proposed framework.

During data analysis, a video frame is divided into multiple blocks of equal size (i.e., 64×64). In each block, the CP-ANN architecture is applied to each pixel to estimate how many pixels in a particular block belong to background scene. This entire procedure can be represented by the following equation.

$$B_q(x_q, y_q) = \begin{cases} 0, & if \quad \sum b(x,y) > 256 \\ 1, & if \quad otherwise \end{cases} , \qquad (15)$$

where q represents ID and $q \in \{1, 2, \cdots, Q\}$, and (x_q, y_q) represents central location of a block in a video frame, respectively. Here, a 0 indicates that most of the pixels in $B_q(x_q, y_q)$ belong to the background scene, and a $B_q(x_q, y_q)$ is classified as a background scene block. On the other hand, a 1 means that the block belongs to a moving object block.

It is possible that in a block classified as a moving object block, some of the pixels may belong to background scene. To locate those pixels, a pixel-based analysis is also performed on those blocks classified as moving object blocks. This analysis is represented by the following equation.

$$\overline{b(x,y)} = \begin{cases} 0, & if \quad B_q(x_q, y_q) > 0.5 \\ 1, & if \quad otherwise \end{cases} . \qquad (16)$$

Here, a 0 means a pixel belongs to background scene while a 1 means it belongs to a moving object. An average threshold value (i.e., 0.5) is selected to obtain a reliable segmentation in this analysis otherwise an extremely lower or higher value may increase false-positive rates.

IV. Experimental Setup and Simulation Results

In this section, we compare the performance of our proposed P2DCA framework with the schemes, i.e., SLICER with Transfer on Meet Up (SLICER-TMU), Minimal Cost Transfer (SLICER-MCT)), and Simple Exchanging (SE), proposed in [33] and [34], respectively. For performance comparison, we consider four different metrics, i.e., computation overhead, communication overhead, reconstruction ratio, and segmentation accuracy.

During simulation, we build a WMSN of 500 nodes out of which only 5% are selected as CHs. This network is setup in Matlab 2017a. These nodes are randomly distributed in an area of $500 \times 500m^2$. Range of each node is set to $100m$. Mobile sinks are constantly moving with a constant speed based at random way-point mobility model [35].

In a cluster-based communication, CHs are responsible for multiple tasks, such as coordinating with BS, managing member nodes, authenticating newly joining nodes, and verifying identities of requesting mobile sinks. In our proposed P2DCA framework, we introduce an extra task for CHs, i.e., data and location aggregation. Unlike location aggregation, data aggregation is a computationally intense task and requires availability of computational and storage resources. Computation overhead metric represents time to aggregate data at CHs alongside performing other tasks. As shown in **Fig. 4**, our proposed P2DCA framework shows better performance as compared to SLICER-TMU and SLICER-MCT. In SE technique, no aggregation is applied on multimedia data, therefore, it shows less computation overhead. In the start of simulations, all schemes show a similar performance. However, the computation overhead remains higher in SLICER-TMU and SLICER-MCT with an increase in data transmitting nodes. We use a lightweight mutual handshaking mechanism to verify identities of newly joining MSNs. To further reduce computational load on CHs, registration of mobile sinks is performed at BS. Once registered, CHs receive information about registered mobile sinks from BS. These features are missing in the targeted schemes.

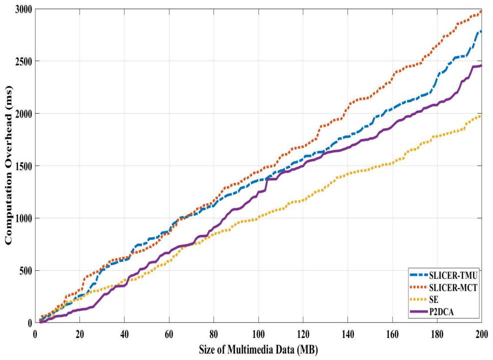

Fig. 4: Computation Overhead

A comparison based on communication overhead is provided in **Fig. 5**. This comparison represents total amount of data that is transmitted from CHs to cloud server via mobile sinks. As shown in this figure, the total communication overhead of our proposed P2DCA framework is lower than other schemes in the presence of an increasing number of data transmitting nodes. Unlike the targeted schemes, our proposed P2DCA framework uses efficient data and location aggregation techniques which reduce the amount of data to be transmitted. Multimedia data coming from the geographical region mostly contain repetitive information. Furthermore, mobile sinks and MSNs operate with limited available bandwidth in open environments. Processing and transmitting of redundant multimedia data require extra computational and storage resources and bandwidth, respectively, which is not suitable for resource and battery limited devices.

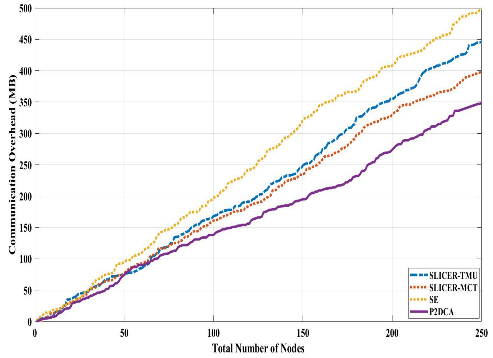

Fig. 5: Communication Overhead

A comparison based on reconstruction ratio is shown in **Fig. 6**. A reconstruction ratio represents the percentage of multimedia data successfully reconstructed by cloud server. As shown in this figure, our proposed P2DCA framework shows better performance in the presence of a packet-drop error of 5%. Unlike the targeted schemes, our proposed framework transmits less amount of data. In the presence of a packet-drop error, missing data packets can be recovered either by retransmition or error concealment. Such a recovery of data packets may introduce excessive delays and increase network load due to retransmission. As a result, the cloud server needs to wait until all data packets are successfully reconstructed/arrived before starting the segmentation process. In real-time IoMT applications, such as video surveillance, healthcare, and transport management, quick actions are required. Due to heavy multimedia data transmission and processing delays, the targeted schemes might not be suitable for real-time IoMT applications.

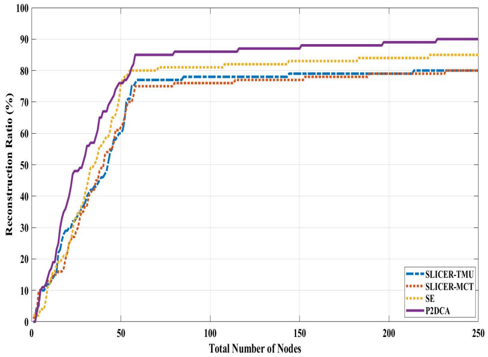

Fig. 6: Reconstruction Ratio

In real-world scenarios, multimedia data is always compressed before transmission. In our experiments, videos are encoded using the Scalable High-efficiency Video Coding (SHVC) standard. Here, we assume that MSNs are fixed nodes. Therefore, selected test video sequences contain moving objects with static backgrounds. Once multimedia data is successfully decoded at cloud server, next task is to perform segmentation to extract meaningful information, e.g., tracking moving objects. We compare performance of our proposed P2DCA framework against the targeted schemes in terms of segmentation accuracy. The segmentation accuracy represents percentage of successful segmentation and reconstruction of meaningful information from reconstructed data. For performance analysis, we use the following metrics, i.e., True-Positive (TP), False-Positive (FP), False-Negative (FN), and F-measure. These are standard metrics used to measure accuracy of binary segmentation. To compute the average performance, we use the following mathematical expressions.

$$\begin{aligned} Recall(Re) &= \frac{\#TP}{\#TP + \#FN}, \\ Precision(Pr) &= \frac{\#TP}{\#TP + \#FP}, \\ F-measure &= \frac{2 \times Pr \times Re}{Pr + Re}, \end{aligned} \tag{17}$$

where, $\#TP$, $\#FN$ and $\#FP$ represent total number of TP, FN, and FP, respectively.

In a quantitative-based evaluation based on Re, Pr and F-measure metrics, a high score means a better performance. An overall quantitative comparison is summarized in **Table I**. As shown, better results in terms of average recall and average precision are obtained on multimedia data processed through our proposed P2DCA framework as compared to the multimedia data processed through the targeted schemes. In an SHVC-based encoding, compression of videos means loss of data and introduction of visual artifacts in videos. As data analysis phase of our proposed framework is running on cloud server, it is important to know that background and foreground segmentation are performed on compressed aggregated videos. Despite this fact, overall accuracy of our proposed framework on video sequences with moving objects is still better than the other schemes as shown in the last column representing average F-measure. Due to no aggregation in SE technique, extensive compression is applied to reduce the size of multimedia data before transmission. High compression and packet-drop error together make it difficult for cloud server to extract meaningful information from received multimedia data, and that is why this technique lacks behind our proposed framework.

Method	Average Re	Average Pr	Average F-Measure
SLICER-TMU	0.7666	0.7643	0.7627
SLICER-MCT	0.7873	0.7973	0.7751
SE	0.7951	0.8451	0.806
P2DCA	0.8151	0.8784	0.8339

TABLE I: Quantitative Comparison

V. Conclusion

In this paper, we have proposed a privacy-preserving-based data collection and analysis framework, called P2DCA for IoMT applications. This framework has been designed to collect multimedia data using mobile sinks from wireless multimedia sensor networks. In our proposed framework, an underlying WMSN has been distributed into small clusters. Before sharing with mobile sinks, data and location information have been aggregated at CHs to hide individual data sources (i.e., member nodes) from mobile sinks. After receiving data from mobile sinks, a cloud server has performed an analysis to segment a video into foreground and background regions to track moving objects. Simulation results have shown that our proposed framework performs better in terms of preserving the privacy of member nodes and segmenting the received data. Regarding future work, we are planning to use the simulation results produced in this paper as a base to perform further enhancements in our proposed framework to support mobile MSNs with random speeds.

References

[1] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, “Internet of multimedia things: Vision and challenges,” *Ad Hoc Networks*, vol. 33, pp. 87–111, 2015.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “Wireless multimedia sensor networks: Applications and testbeds,” *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, 2008.

[3] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[4] K. Yang, K. Zhang, J. Ren, and X. Shen, “Security and privacy in mobile crowdsourcing networks: challenges and opportunities,” *IEEE communications magazine*, vol. 53, no. 8, pp. 75–81, 2015.

[5] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, “Intrusion detection by machine learning: A review,” *Expert Systems with Applications*, vol. 36, no. 10, pp. 11 994–12 000, 2009.

[6] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 439–450.

[7] Y. Lindell and B. Pinkas, “Privacy preserving data mining,” in *Annual International Cryptology Conference*. Springer, 2000, pp. 36–54.

[8] ——, “Privacy preserving data mining.” *Journal of cryptology*, vol. 15, no. 3, 2002.

[9] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, “Adversarial machine learning,” in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. ACM, 2011, pp. 43–58.

[10] N. M. Oliver, B. Rosario, and A. P. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[11] A. Pentland and A. Liu, “Modeling and prediction of human behavior,” *Neural computation*, vol. 11, no. 1, pp. 229–242, 1999.

[12] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, “A security and privacy review of vanets,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, 2015.

[13] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, “Security for 4g and 5g cellular networks: A survey of existing authentication and privacy-preserving schemes,” *Journal of Network and Computer Applications*, 2017.

[14] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, “Privacy-preserving fusion of iot and big data for e-health,” *Future Generation Computer Systems*, 2018.

[15] L. Zhang, J. Song, and J. Pan, “A privacy-preserving and secure framework for opportunistic routing in dtns,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7684–7697, 2016.

[16] Q. Kong, R. Lu, M. Ma, and H. Bao, “A privacy-preserving sensory data sharing scheme in internet of vehicles,” *Future Generation Computer Systems*, 2017.

[17] Z. Yan, W. Ding, V. Niemi, and A. V. Vasilakos, “Two schemes of privacy-preserving trust evaluation,” *Future Generation Computer Systems*, vol. 62, pp. 175–189, 2016.

[18] K. Xie, X. Ning, X. Wang, S. He, Z. Ning, X. Liu, J. Wen, and Z. Qin, “An efficient privacy-preserving compressive data gathering scheme in wsns,” *Information Sciences*, vol. 390, pp. 82–94, 2017.

[19] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, “Private machine learning classification based on fully homomorphic encryption,” *IEEE Transactions on Emerging Topics in Computing*, 2018.

[20] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, “Differentially private naive bayes learning over multiple data sources,” *Information Sciences*, 2018.

[21] C.-z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, “Privacy-preserving naive bayes classifiers secure against the substitution-then-comparison attack,” *Information Sciences*, 2018.

[22] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, “Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud,” *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 467–479, 2014.

[23] T. Zhang and Q. Zhu, “Dynamic differential privacy for admm-based distributed classification learning,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, 2017.

[24] ——, “Distributed privacy-preserving collaborative intrusion detection systems for vanets,” *IEEE Transactions on Signal and Information Processing over Networks*, 2018.

[25] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, “Multi-key privacy-preserving deep learning in cloud computing,” *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[26] Y. Shen, C. Luo, D. Yin, H. Wen, R. Daniela, and W. Hu, “Privacy-preserving sparse representation classification in cloud-enabled mobile applications,” *Computer Networks*, vol. 133, pp. 59–72, 2018.

[27] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[28] M. Usman, M. A. Jan, X. He, and P. Nanda, “Data sharing in secure multimedia wireless sensor networks,” in *Trustcom/BigDataSE/I SPA, 2016 IEEE*. IEEE, 2016, pp. 590–597.

[29] M. Jan, P. Nanda, M. Usman, and X. He, “Pawn: a payload-based mutual authentication scheme for wireless sensor networks,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 17, 2017.

[30] M. A. Jan, F. Khan, M. Alam, and M. Usman, “A payload-based mutual authentication scheme for internet of things,” *Future Generation Computer Systems*, 2017.

[31] F. Zhang, R. Safavi-Naini, and W. Susilo, “An efficient signature scheme from bilinear pairings and its applications,” in *International Workshop on Public Key Cryptography*. Springer, 2004, pp. 277–290.

[32] D. Graupe, *Principles of artificial neural networks*. World Scientific, 2013, vol. 7.

[33] F. Qiu, F. Wu, and G. Chen, “Privacy and quality preserving multimedia data aggregation for participatory sensing systems,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1287–1300, 2015.

[34] D. Christin, J. Guillemet, A. Reinhardt, M. Hollick, and S. S. Kanhere, “Privacy-preserving collaborative path hiding for participatory sensing applications,” in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 341–350.

[35] C. Bettstetter, G. Resta, and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” *IEEE Transactions on mobile computing*, vol. 2, no. 3, pp. 257–269, 2003.