Usman, M., Jolfaei, A., & Jan, M. A. (2020). RaSEC: An Intelligent Framework for Reliable and Secure Multilevel Edge Computing in Industrial Environments. *IEEE Transactions on Industry Applications*, *56*(4), 4543–4551.

https://doi.org/10.1109/TIA.2020.2975488

RaSEC: An Intelligent Framework for Reliable and Secure Multi-Level Edge Computing in Industrial Environments

Muhammad Usman, *Member, IEEE,* Alireza Jolfaei*, *Senior Member, IEEE,* and Mian Ahmad Jan*

***Abstract*—Industrial applications generate big data with redundant information that are transmitted over heterogeneous networks. The transmission of big data with redundant information not only increases the overall end-to-end delay but also increases the computational load on servers which affects the performance of industrial applications. To address these challenges, we propose an intelligent framework for Reliable and Secure multi-level Edge Computing (RaSEC) in industrial environments. This framework operates in three phases. In the first phase, level-one edge devices apply a lightweight aggregation technique on the generated data. This technique not only reduces the size of the generated data, but also helps in preserving the privacy of data sources. In the second phase, a multi-step process is used to register Level-Two Edge Devices (LTEDs) with High-Level Edge Devices (HLEDs). Due to the registration process, only legitimate LTEDs can forward data to the HLEDs, and as a result, the computational load on HLEDs decreases. In the third phase, the HLEDs use a convolutional neural network to detect the presence of moving objects in the data forwarded by LTEDs. If a movement is detected, the data are uploaded to the cloud servers for further analysis otherwise the data are discarded which minimizes the use of computational resources on cloud computing platforms. Simulation results show that our proposed framework is highly resilient against security and privacy threats. The proposed framework also helps in increasing the response time by forwarding useful information to the cloud servers and can be utilized by various industrial applications.**

***Index Terms*—intelligent, secure, edge computing, privacy, convolutional neural network.**

I. Introduction

Industrial applications like healthcare, transportation management, and surveillance, generate huge volumes of data that are sent to the cloud servers for processing and storage. The data are always forwarded to the cloud servers over public networks, e.g., the Internet, based on best-effort delivery service. The cloud computing platforms offer ample of computing and storage resources, however, they are unable to deal with critical issues like response time, security and privacy, in the underlying networks. To deal with these issues, a Multi-Level Edge Computing (MLEC) architecture can be utilized [1]. In this architecture, different computing tasks are set for geographically distributed edge devices. The MLEC architecture helps in bringing the computational resources closer to data sources, e.g., end-devices and applications, in industrial environments. Although the MLEC architecture helps in distributing the processing load between edge devices, challenges like security, privacy and response time, still need to be addressed.

In the traditional MLEC architecture, the computational resources are usually owned by different providers, e.g., Internet service providers, cloud service providers, and enterprise organizations that own a complete network and a cluster setup [2], [3]. Due to multiple ownership of computing resources in the MLEC architecture, a malicious device can easily enter the network, act like a legitimate edge device, and connect with end-devices and industrial applications to control them. Due to the absence of a centralized authentication mechanism in the MLEC architecture, malicious edge devices may get access to data generated by end-devices and sensitive industrial applications, e.g., healthcare, surveillance, and transportation management, and manipulate and misuse it or forward it to an unknown destination. In another example, a malicious end-device or application may try to connect to a legitimate edge device and requests for access to sensitive data stored on it or forwards fake data to keep it occupied. In either scenario, there is a need for a device authentication mechanism in the MLEC architecture to avoid the misuse of available computing resources.

In the MLEC architecture, the edge devices are in the vicinity of end-devices. A compromised edge device can collect sensitive information, e.g., location coordinates information and identities (IDs), of end-devices and use it for malicious purposes. To protect the privacy of end-devices, various privacy-preserving techniques have been introduced in past few years [4]. These techniques are basically designed for cloud computing based architectures. In these techniques, homomorphic encryption and differential privacy algorithms are used to preserve the privacy of data sources and allow to search a specific type of data through keywords without decryption. Although these techniques are efficient, they are computationally complex, and are not feasible for industrial applications using MLEC architecture. The industrial applications, e.g., surveillance, transportation management, and healthcare, generate huge volumes of hybrid data. Applying computationally complex privacy-preserving techniques on generated data may increase the processing time which may affect the performance of aforementioned industrial applications. Therefore, there is a need for a lightweight and efficient privacy-preserving

The asterisk indicates corresponding authors.

Muhammad Usman is the School of Science, Engineering, and Information Technology, Federation University, Australia. (E-mail: muhammad.usmanskk@gmail.com)

Alireza Jolfaei is with the Department of Computing, Macquarie University, Australia (E-mail: alireza.jolfaei@mq.edu.au)

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

technique that can be applied on multimedia data and protect the privacy of end-devices and industrial applications based on MLEC architecture.

Data generated by industrial applications are always sent to cloud servers for storage and analysis. However, it is possible that the applications may generate redundant data, and as a result, the cloud sources may be misused. Furthermore, transmission of redundant data over public networks, e.g., the Internet, may increase the latency and response time which may affect the performance of real-time industrial applications. To address the latency and response time issues, researchers have proposed various intelligent edge computing based approaches over the past few years [5]. In the intelligent edge computing based approaches, lightweight machine learning algorithms are implemented on edge devices to reduce the amount of data sent to the cloud servers. The intelligent edge devices can help in decreasing the network latency and improving the response time of real-time industrial applications. However, the machine learning algorithms are computationally complex, and take significantly a large amount of time to process the input data. Therefore, there is a need for a lightweight machine learning algorithm on edge devices that can quickly process the input data and decide whether the data need to be forwarded to the cloud servers or not.

In this paper, we propose an intelligent framework for Reliable and Secure multi-level Edge Computing (RaSEC) in industrial environment. This framework operates in three phases to address security, privacy, and response time issues. In the first phase, a lightweight aggregation technique is used by Level-One Edge Devices (LOEDs) to minimize the size of multimedia data and preserve the privacy of end-devices. In the second phase, Level-Two Edge Devices (LTEDs) are registered with High-Level Edge Devices (HLEDs) using a handshaking mechanism to protect the framework from malicious edge devices. In the last phase, the HLEDs use Convolutional Neural Network (CNN) to process the aggregated data to detect the redundancy on an advanced level. The major contributions of our proposed framework are as follows.

- To the best of our knowledge, our proposed framework is the first effort to deal with security, privacy and response time issues using a MLEC architecture. This framework is designed to process big multimedia data generated by real-time industrial applications.
- A lightweight aggregation technique is applied to minimize the size of generated multimedia data. The redundancy in the generated data is detected through a similarity-based comparison between video frames. The location coordinates information of end-devices is also aggregated to protect their privacy by hiding their actual location coordinates information.
- An energy-efficient technique is used to register the LTEDs with the HLEDs. Multiple encrypted messages are exchanged between the LTEDs and HLEDs to complete the registration process. Once the registration process is completed, the LTEDs are allowed to forward the collected data to HLEDs.
- A lightweight CNN-based architecture is implemented on HLEDs to analyze the aggregated data forwarded by LTEDs using multiple connected and hidden layers to extract features from input data and detect the movements of objects. Based on the analysis, the HLEDs decide whether the aggregated data need to be discarded or forwarded to the cloud servers for processing and storage.

The rest of this paper is organized as follows. An overview of recent efforts on the security, privacy, and response time issues in the edge computing domain is provided in Section II. The proposed framework is discussed in detail in Section III. Simulation results are explained in Section IV. Finally, the paper is concluded in Section V.

II. Literature Review

This section provides an overview of recent developments made in the edge computing domain. In the following subsections, we discuss various approaches that are proposed to deal with issues like security, privacy, and data management using machine learning in the edge computing architecture.

A. Security Efforts

A survey on research challenges in connecting edge and cloud computing architectures was presented in [6]. In this survey, the research challenges focus on security and forensic issues when achieving a higher throughput using concurrent access and mobility support. Another survey on extending cloud services using mobile edge and fog computing architectures was presented in [7]. This survey highlights various security and resilience issues when using cloud computing services over public networks. A survey on security issues in Internet of Things (IoT) based architectures supported by mobile edge computing was presented in [8]. This survey focuses on the use of mobile edge computing to resolve security issues in certain IoT applications, e.g., environment perception and vehicular networks. Another survey on security and privacy issues in edge computing paradigm was presented in [9]. This survey analyzes existing security and privacy attacks and countermeasures provided by the traditional edge computing paradigm. Surveys presented in [6]–[9] highlight various security issues in edge computing based architectures. However, these surveys do not highlight security issues when streaming multimedia data generated by real-time industrial applications.

A secured IoT service architecture, based on edge and cloud computing, was presented in [10]. In this architecture, a trust evaluation mechanism is introduced to reduce resource consumption and improve IoT-cloud services. A secured pairing-based key agreement protocol for smart grids based on the edge computing architecture was proposed in [11]. This protocol provides an anonymous and secured authentication facility between utility control and smart meter without involving a third party. A two-layer detection engine with hybrid feature analysis for edge computing paradigm was proposed in [12]. This engine detects various mobile IoT malware as compared to the existing malware detection engines. Approaches presented in [10]–[12] provide security services in the edge computing based environments. However, these approaches are computationally complex, and may not be feasible for industrial applications generating real-time multimedia data.

B. Privacy Efforts

A survey on security and privacy challenges in the edge computing architecture was presented in [13]. In this survey, various security and privacy concerns are reviewed, and ongoing research efforts and future research directions are highlighted. A survey of different IoT-related architectures was presented in [14]. In this survey, various technologies and applications, which can help in addressing security and privacy challenges in IoT-related architectures are discussed. Various privacy-preserving issues in querying IoT-related architectures were surveyed in [15]. This survey focuses on recent privacy-preserving approaches along with their pros and cons. Surveys presented in [13]–[15] discuss various privacy-related issues in the edge computing architecture. However, these surveys do not provide a discussion on privacy-preserving issues during multimedia streaming in the edge computing architecture.

Privacy-preserving schemes for mobile edge computing applications were proposed in [16], [17]. These schemes encrypt data to preserve the privacy of end-devices. A privacy-preserving scheme based on differential privacy was proposed in [18]. In this scheme, the Laplacian mechanism is applied to unknown attributes of collected data to protect the privacy. A privacy-preserving framework based on local differential privacy was proposed in [19]. In this framework, data are aggregated and distilled at network edge to preserve the privacy of users' sensitive information. A privacy-preserving approach for mobile edge clouds was proposed in [20]. In this approach, heuristics strategies are adopted at network edge to distract eavesdroppers by mimicking users' mobility and minimizing users' tracking accuracy. Approaches presented in [16]–[20] use computationally complex processes to preserve the privacy of end-users and applications, and may not be feasible for real-time edge computing based applications operating in industrial environment.

C. Machine Learning Efforts

A survey on the use of machine learning techniques in the edge computing architecture was presented in [21]. This survey reviews different machine learning techniques that can be used to analyze data streams generated by various industrial applications. Surveys on machine learning algorithms for IoT-based architectures and wireless sensor networks were presented in [22]. In these surveys, various machine learning algorithms are studied to analyze the generated data in the IoT-based architectures and wireless sensor networks. These surveys describe various challenges, such as sink scheduling, congestion control, synchronization between services, energy harvesting, and response time. Surveys presented in [21], [22] discuss various machine learning algorithms to deal with different time critical factors in IoT and edge computing architectures, however, they do not provide any discussion on real-time multimedia data processing.

Personalized healthcare systems, based on cloud and edge-enabled network, were presented in [23], [24]. These systems use edge computing and deep learning technologies to process big healthcare data in real-time. A deep learning and edge computing based system for food recognition was proposed in [25]. This system uses deep learning based visual food recognition algorithms to overcome latency and battery lifetime issues faced by end-devices. Systems proposed in [23]–[25] use deep learning techniques to deal with latency and response time challenges faced by IoT-based applications. However, they may not be feasible for real-time industrial applications.

III. Reliable and Secure Multi-Level Edge Computing

In this section, we explain our proposed RaSEC framework. A block diagram of our proposed framework is shown in Fig. 1. Our proposed framework operates in three phases. In the first phase, multimedia data generated by industrial applications are forwarded to LOEDs as shown in Fig. 1.

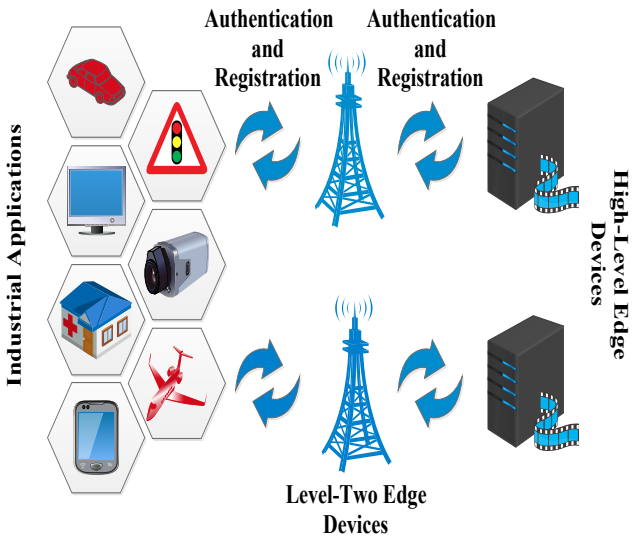

Fig. 1. RaSEC Framework

The industrial applications are using Multimedia Sensor Nodes (MSNs) to generate the data. These nodes build up a network, called Wireless Multimedia Sensor Network (WMSN). After receiving the data, the LOEDs apply a lightweight aggregation technique to reduce the size of data and preserve the privacy of end-devices by hiding their IDs and location coordinates information. The LOEDs forward aggregated data to nearby LTEDs. The LTEDs are responsible to collect data from LOEDs and forward to the HLEDs without revealing information of LOEDs. In the second phase, the LTEDs need to register with HLEDs before forwarding the collected data. The registration process is based on a mutual handshaking mechanism and completed in five steps. In the last phase, after receiving the data from registered LTEDs, the HLEDs use a CNN-based architecture to detect the movements of objects. If a movement is detected, the data are forwarded to cloud servers for further analysis. In the following subsections, we explain these phases in detail. Symbols used in the following subsections are summarized in Table I.

Notation	Description
γ	Similarity index value
H_s,H_r	Histograms
β	LTED
R	Registration request
μ	Session key
η	Random nonce
C	Challenge
Δ	Time-stamp
δ	Authentication value
$\overline{R}$	Response
Ω	Registration certificate
F	Function
f_1, f_2	Functions to motion detection
α_1, α_2	Predefined weights
s	Data sample
I	Total data samples
K	Total predefined motions

TABLE I. Notations of RaSEC framework

A. Aggregation Phase

In this phase, two tasks are performed, i.e., distributing an underlying WMSN into multiple clusters and aggregating the data. These two tasks are performed to manage the network and generated data, and preserve the privacy of data sources, i.e., MSNs.

In the first task, the WMSN is partitioned into multiple clusters. Each cluster consists of N nodes. Out of these N nodes, only one node is selected as a LOED to manage and represent the cluster. The selected LOEDs are responsible to collect data from member MSNs and coordinate with the LTEDs. The first task, i.e., partitioning the WMSN and selecting the LOEDs, is based on our previous works published in [26]–[28].

The second task deals with captured multimedia data. In our proposed framework, we consider videos as the multimedia data. In practical scenarios, the MSNs are randomly deployed. Due to random deployment, it is possible that multiple MSNs may capture the same scene but at different angles. As a result, videos with similar visual contents are forwarded to LOEDs which creates the redundancy issue. Redundant video data not only require extra computational resources, but also require extra bandwidth to transmit from LOEDs to HLEDs via LTEDs. Furthermore, if the redundant video data are leaked during transmission, it may reveal information of data sources, and as a result, the privacy of MSNs can easily be compromised. To address these two challenges, i.e., data redundancy and privacy protection, the LOEDs aggregate the received video data and location coordinates information of member MSNs. The aggregation of location coordinates information is a simple process and does not require computational resources. In this process, the LOEDs compute a mean value of location coordinates of member MSNs in a cluster and use the computed mean value as location coordinates of aggregated data. From aggregated location coordinates information, it is hard to locate the actual location coordinates values of individual MSNs, and as a result, their privacy is preserved.

To minimize the redundancy in captured videos, the LOEDs uses a similarity-based technique. It is a lightweight technique that distributes the videos into two data sets, i.e., standard and redundant data sets. In cluster-based communication, the LOEDs also captures videos. Videos captured by LOEDs are placed into the standard data set while videos captured by member MSNs are placed in the redundant data set. A video is a combination of multiple pictures called frames. In video processing, the frames are distributed into multiple groups where the sizes of groups can be variable or fixed, depending upon the availability of buffer space and computational power. In our proposed framework, the videos in each data set are distributed into multiple groups of frames of same size, i.e., 10 video frames per group, and the comparison is performed on a frame-by-frame basis as shown in Fig. 2.

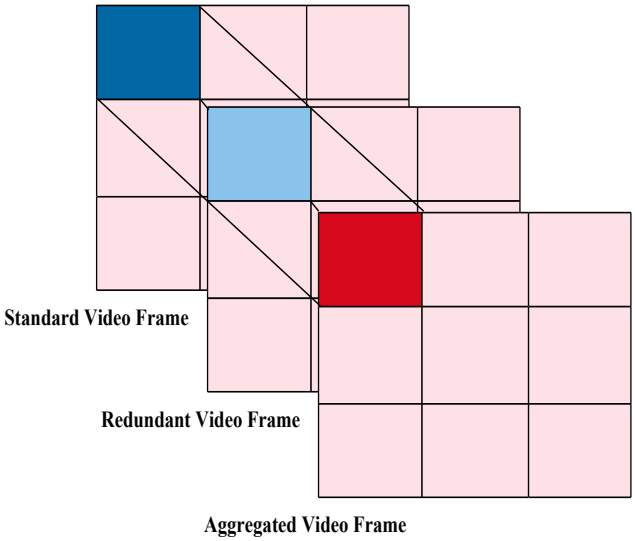

Fig. 2. Frame Comparison

In the comparison process, a video frame can be partitioned into multiple blocks of different shapes and sizes. Unlike the fixed shape and size, the variable shape and size can produce high accuracy at the cost of computational complexity. The variable shape and size based processing is useful for applications like tracking and identification of multiple objects moving with variable speeds. However, in real-life scenarios, it is possible that the MSNs may not always be capturing videos containing multiple objects moving with variable speeds. To reduce the computational complexity and processing load on LOEDs, our proposed framework partitions video frames into square-shaped blocks of equal size, i.e., 64×64. After partitioning the video frames, a histogram normalization technique is applied on each block to compute a similarity index value (i.e., γ) as shown in the following equation.

$$\gamma = H_s \times log_2\left(\frac{H_s}{H_r}\right), \tag{1}$$

where H_s and H_r represent histograms of blocks of video frames from standard and redundant data sets, respectively.

After computing the similarity index value, it is compared against a predefined threshold (i.e., $\overline{\gamma}$), where $\overline{\gamma} \in (\gamma_{min}, \gamma_{max})$. If the computed value is less than the predefined threshold, then the block is deemed to be dissimilar from the corresponding block selected from standard data set. If 25% or more blocks in a selected video frame from redundant data set are found dissimilar, the frame is considered different and is kept in the buffer. Once the comparisons of frames from both data sets are completed, the LOEDs create aggregated videos consisting of all dissimilar frames. After aggregating the video data, the LOEDs associate it with the computed mean values of location coordinates information and forward them to LTEDs.

B. Registration Phase

To securely transmit data to HLEDs, the LTEDs need to get registered. The registration is performed in five steps, i.e., Registration request, Challenge to request, Response to challenge, Registration certificate, and Share databases of LOEDs.

In the first step, an LTED (i.e., β_j, $j \in \{1, 2, \cdots, J\}$) sends a registration request to a nearby HLED. It is possible that the HLED may be dealing with multiple LTEDs. Therefore, the requesting LTED creates an encrypted registration request (i.e., R_j) by using the following equation.

$$R_j = \text{AES}\{j, (\eta_j \oplus \mu_j)\}, \tag{2}$$

where μ_j and η_j represent a session key and a random nonce, respectively, each having 128-bit length and being unique to the requesting LTED. The AES-128 bit in CBC mode (Cipher Block Chaining) is used for encrypting messages between nodes. It is chosen due to its smaller computational and storage overhead as compared to other popular algorithms like Rivest–Shamir–Adleman (RSA) and Elliptic-Curve Cryptography (ECC).

After receiving the R_j, the HLED decrypts and retrieves the embedded values. In the second step, the HLED uses the retrieved values to create an encrypted challenge for the requesting LTED. The encrypted challenge (i.e., C_j) is created using the following equation.

$$\Delta_j = (j \mid\mid \eta_{e_1}), \tag{3a}$$

$$\delta_j = (j \mid\mid \eta_j \mid\mid \eta_{e_1}), \tag{3b}$$

$$C_j = AES\{\Delta_j, \delta_j\}, \tag{3c}$$

where Δ_j and δ_j represent a time-stamp and an authentication value, respectively, and η_{e_1} is a random nonce generated by HLED. The time-stamp represents the total amount of session time between the requesting LTED and HLED, and the authentication value is used to create a response to the challenge generated by HLED.

After receiving the encrypted challenge from HLED, the LTED decrypts and retrieves the embedded values. In the third step, the LTED creates an encrypted response (i.e., $\bar{R}_j$) using the forwarded δ_j and η_{e_1} as shown in the following equation.

$$\bar{R}_j = AES\{\eta_j, (\delta_j \mid\mid (\eta_j \oplus \mu_j) \mid\mid \eta_{e_1})\}. \tag{4}$$

After receiving the encrypted response, the HLED verifies the identity of the requesting LTED by retrieving δ_j and η_{e_1}. If the values exist, the LTED is considered authentic. In the fourth step, the HLED generates a registration certificate (i.e., Ω_j) by using the following equation.

$$\Omega_j = \eta_{e_2}.(j \mid\mid (\eta_j \oplus \mu_j)), \tag{5}$$

where η_{e_2} is another random nonce generated by HLED.

After generating the registration certificate, the HLED makes two copies of it. The first copy is sent to the requesting LTED as an acknowledgment and the second copy is stored at HLED for future communication. In the last step, the LTED forwards the first batch of aggregated data upon receiving the registration certificate, to the HLED. This entire registration process is summarized in Algorithm 1.

Algorithm 1 LTED Registration

Initialization:
- $\beta_j \longleftarrow j$, where $j \in \{1, 2, 3, ..., J\}$
- $\beta_j \longleftarrow [\mu_j, \eta_j]$
- HLED $\longleftarrow [\eta_{e_1}, \eta_{e_2}]$
- Input $[j = 2^{128}, \mu_j = 2^{128}, \eta_j = \eta_{e_1} = \eta_{e_2} = 2^{128}]$

$\beta_j \rightarrow$ HLED : $\{R_j$, Registration request$\}$
HLED $\rightarrow \beta_j$: $\{C_j$, Challenge to request$\}$
$\beta_j \rightarrow$ HLED : $\{\bar{R}_j$, Response to challenge$\}$
if (δ_j, η_{e_1}) exist **then**
    β_j is authentic
    HLED $\rightarrow \beta_j$: $\{\Omega_j$, Registration certificate$\}$
else
    β_j is unauthentic
end if

C. Analysis Phase

In this phase, we use a network of layers and modules based on CNNs to analyze the aggregated videos and detect the movements of objects. This network consists of four modules and two ordinary convolutional layers. The first two modules are called convolutional modules, and the third and the fourth modules are called detection and regression modules, respectively. The convolutional modules consist of two variable-sized convolutional kernels. These kernels are used to extract different features from input data on different scales. They can also help in detecting objects of different sizes. The detection and regression modules consist of two fully connected layers to analyze the features that are extracted and forwarded by the lower layers. The output of the detection module varies between 0 and 1 while the regression module is used to predict different characteristics of the detected objects. This architecture is based on a work proposed in [29]. It was used to analyze and understand a scene. Based on this work, we propose a function (i.e., F) that performs two tasks, i.e., detecting a motion and determining the degree of the motion, and is represented by the following equation.

$$F = \alpha_1 f_1 + \alpha_2 f_2, \tag{6}$$

where f_1 and f_2 are functions to detect a motion and the degree of a motion, respectively, and α_1 and α_2 are predefined weights to adjust relative performances of f_1 and f_2, respectively.

The function f_1 is based on the Softmax loss function, and is used to detect the motion of an object in a specific region. If an input data sample is represented by s_i, and its ground truth label by l_i, for $i = 1, 2, \cdots, I$, where I represents the total number of input samples, then the s_i can be analyzed by function f_1 as shown in the following equation.

$$f_1 = -\frac{\sum_{i=1}^{I}\sum_{k=1}^{K} 1\{l_i = \hat{l}_k\} log\left\{\frac{e^{m_1^{(k)}(s_i)}}{\sum_{k=1}^{K} e^{m_1^{(k)}(s_i)}}\right\}}{I}, \tag{7}$$

where K represents the total number of predefined motions, $m_1^k(s_i)$ represents the corresponding output of the Softmax classification layer, $\hat{l_k}$ is a real value for possible results, and $e^{m_1^k(s_i)}$ represents the possibility of a specific type of motion and varies between 0 and 1.

The function f_2 is a regression function used to define the degree of motion. This function is based on a bounding box regression method proposed in [30]. It is a popular technique to refine or predict localization boxes in object detection approaches. Typically, bounding-box regressors are trained to regress from either region proposals or fixed anchor boxes to nearby bounding boxes of a pre-defined target object classes. In our proposed framework, the function f_2 is expressed in the following equation.

$$f_2 = \frac{\sum_{i=1}^{I} \text{smooth}_{L_1}(m_2^{(d_i)}(s_i) - d_i)}{I}, \tag{8a}$$

$$\text{smooth}_{L_1}(c) = \begin{cases} 0.5c^2, & \text{if } |c| < 0 \\ |c| - 0.5, & \text{otherwise} \end{cases}, \tag{8b}$$

where $m_2^{(d_i)}(s_i)$ represents the output of the Smooth_{L_1} localization layer and d_i represents the central coordinates, dimensions and Euler angles of moving objects.

The HLEDs analyze the aggregated videos forwarded from LTEDs using the aforementioned CNN-based network and make the following decisions.

- If no motion is detected, the aggregated videos are discarded by assuming that the data do not contain any useful information and do not need to be forwarded to the cloud servers.
- If a motion is detected, then some actions need to be taken by HLEDs.
  - Priorities need to be set for data coming from different LTEDs.
  - The LTEDs forwarding multimedia data containing the highest degree of motion need to be given the highest priority and their data need to be processed first at the HLEDs. The multimedia data containing the highest degree of motion need to be offloaded immediately and sent to the cloud servers for a detailed analysis.
  - If the multimedia data contain a small degree of motion, the HLEDs need to wait for more data to arrive from LTEDs to analyze whether the degree of motion is consistent or keeps increasing. In either case, the multimedia data need to be offloaded to the cloud servers for further analysis.

IV. Experimental Setup

In this section, we evaluate the performance of our proposed framework. This framework is designed to deal with three challenges, i.e., privacy, security, and response time. Metrics like computational and transmission overheads are used to evaluate the performance of privacy and security features of our proposed framework. The resilience of security feature is also tested against well-known attacks. The performance of analysis feature is evaluated in terms of detection accuracy. For simulations, we use Matlab 2018a. The proposed framework consists of 500 randomly deployed MSNs. In each round of simulation, only 5% MSNs are selected as LOEDs.

Our proposed framework is compared with existing privacy-preserving frameworks, i.e., SLICER with Transfer on Meet Up (SLICER-TMU) and Minimal Cost Transfer (SLICER-MCT) [31]. These two frameworks use k anonymity rule for privacy preservation. The privacy is preserved through data coding techniques and message transfer strategies. Similar to our proposed framework, these frameworks use multiple algorithms to achieve the goal. We also compare the performance with another framework, called Simple Exchanging (SE) [32]. In this framework, the privacy of location coordinates is preserved by applying a decentralized mechanism, i.e., aggregation technique. Although, these frameworks provide privacy-preserving services, they have certain limitations. Firstly, they can only protect the privacy of location coordinates of data sources. Secondly, they are designed for small-scale networks. Lastly, their is no support to preserve the privacy in end-to-end communication. On the other hand, our proposed framework covers all these limitations as shown in simulation results in this section.

In the MLEC architecture, edge devices on each level are responsible for multiple tasks. The LOEDs are responsible for verifying and authenticating newly joining MSNs and managing existing MSNs. They are also responsible to protect the privacy of member MSNs. To protect the privacy, the LOEDs use an aggregation technique. Unlike the location coordinates aggregation, the data aggregation task becomes computationally intensive if it is performed on multimedia data, e.g., videos. To aggregate multimedia data, the LOEDs require computational resources and buffer storage to process the forwarded videos on frame-by-frame bases. Fig. 3 shows the performance of our proposed framework in terms of computational overhead. As shown in Fig. 3, our proposed framework has a lower computational overhead as compared to the SLICER-TMU and SLICER-MCT frameworks, but has a slightly higher computational overhead than the SE framework due to the absence of aggregation process.

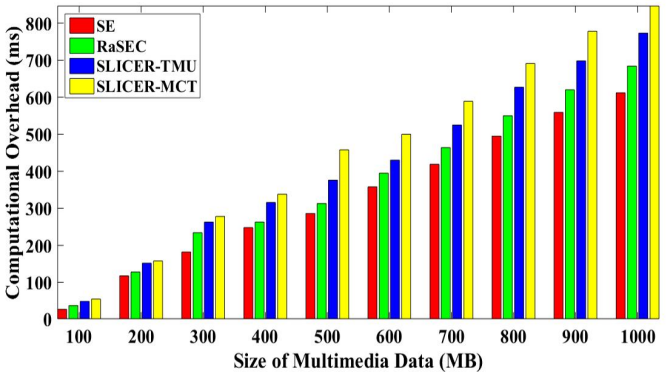

Fig. 3. Computational Overhead

Data aggregation not only helps in preserving the privacy of member MSNs, but also helps in reducing the size of generated data. In the case of multimedia data, sufficient amount of bandwidth is required, and as a result, the underlying WMSN may face problems like heavy network traffic, latency, and end-to-end delay. These problems ultimately affect the response time of applications operating in industrial environments. As shown in Fig. 4, the SE framework generates a huge amount of network traffic which ultimately increases the transmission overhead. On the other hand, our proposed framework shows the least transmission overhead and outperforms the targeted frameworks.

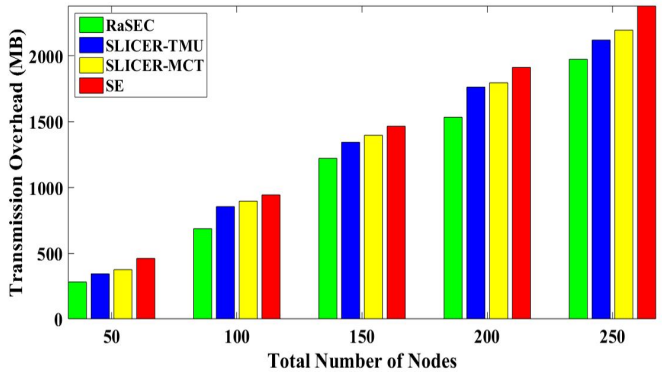

Fig. 4. Transmission Overhead

In Fig. 5, we evaluate data freshness in the presence of up to 10 malicious nodes. A high percentage of data freshness means that a scheme is highly robust against replay attacks. We consider malicious nodes as those entities that infiltrate a network by registering with the HLEDs. Our proposed algorithm, i.e., Algorithm 1, is highly resilient against replay attacks and ensures that the freshness of data is maintained all the time. In the presence of 10 malicious nodes, the average data freshness of our proposed framework is 98.15%, the SLICER-TMU framework is 96.3%, the SLICER-MCT is 95.5%, and the SE framework is 92.01%, over a period of 120 minutes. Our proposed framework achieves the highest percentage of data freshness and hence sustains itself against replay attacks over the course of time. Unlike our proposed framework, the Slicer-TMU and Slicer-MCT frameworks experience a significant decrease in data freshness after 45 minutes of network deployment. The SE framework, on the other hand, experiences a significant drop of 7.5% after the initial 20 minutes of network deployment. Unlike the targeted frameworks, our proposed framework is resilient against replay attacks and is marginally prone to such attacks. The proposed algorithm 1 is lightweight. It uses 128-bit encryption and consumes less energy at LTEDs. It is worth to note that the presence of 256-bit encryption can further enhance the data freshness and ensure a further reduction of replay attacks.

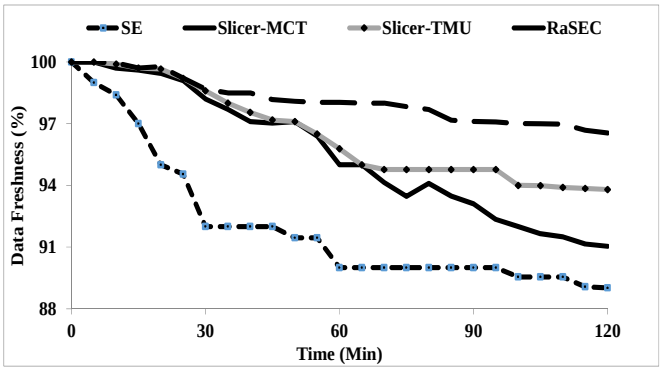

Fig. 5. Data Freshness

In Fig 6, packet delivery ratio is shown for a varying number of malicious nodes. The average packet delivery ratio for our proposed framework is 94.96%, the SLICER-TMU framework is 92.23%, the SLICER-MCT framework is 85.31%, and the SE framework is 74.1%. Unlike our proposed framework, the targeted frameworks experience significant drops in the packet delivery ratios as the number of malicious nodes increases. In general, this metric is associated with the authentication rate. During the authentication process, the malicious LTEDs that go undetected forward their malicious and fabricated data packets to HLEDs. Due to this forwarding, the legitimate LTEDs are unable to transmit their data and they need to wait for longer times. This longer wait may result in buffer overloading, which ultimately leads to packet-drop and a loss of valuable information.

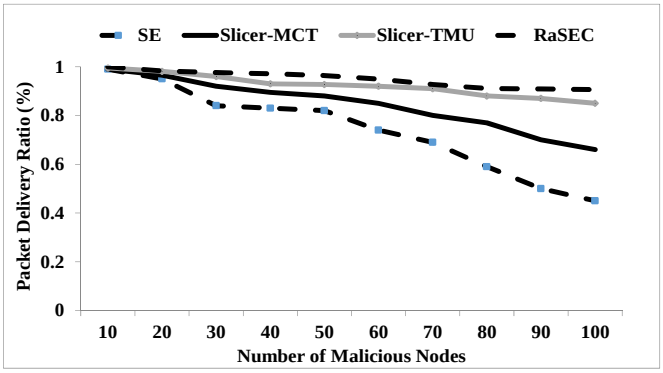

Fig. 6. Packet Delivery Ratio

Videos processed through our proposed framework and the targeted frameworks are analyzed on HLEDs to detect the

movements of objects. In our proposed framework, we assume that the member MSNs are fixed nodes and capture videos with static backgrounds. To perform a comparison, we use different metrics, i.e., Recall (Re), Precision (Pr), and F-measure. The F-measure is approximately the average of the precision and recall when they are close, and is more generally the harmonic mean, which, for the case of two numbers, coincides with the square of the geometric mean divided by the arithmetic mean. It is also known as F1 measure because recall and precision are evenly weighted. These metrics are commonly used in the machine learning domain to perform a quantitative-based evaluation and test the accuracy. These metrics are expressed in the following equations.

$$\text{Re} = \frac{\#tp}{\#tp + \#fn}, \tag{9a}$$

$$\text{Pr} = \frac{\#tp}{\#tp + \#fp}, \tag{9b}$$

$$\text{F-measure} = \frac{2 \times \text{Pr} \times \text{Re}}{\text{Pr} + \text{Re}}, \tag{9c}$$

where $\#tp$, $\#fn$ and $\#fp$ represent the total number of true-positives, false-negatives and false-positives, respectively.

Multimedia data are always compressed at sources before transmission and decompressed on destinations. Here, we consider the LTEDs as sources and the HLEDs as destination. The reconstruction quality depends on various factors, e.g., compression technique, quantization parameters, packet-drop during transmission, and resolution of videos. In this work, the objects and their movements are detected in compressed and aggregated videos. A comparison based on the Re, Pr, and F-measure is shown in Table II. This comparison is performed on video data generated by our proposed framework and the targeted frameworks. It can be seen in Table II that our proposed framework shows a better performance and achieves the highest scores as compared to the targeted frameworks. The SE framework does not aggregate multimedia data, and as a result, extensive compression are applied to multimedia data to reduce the size. Higher compression ratios and packet-drop errors make the detection task difficult. Therefore, the lowest performance is observed in the case of SE framework.

Method	**Average Re**	**Average Pr**	**Average F-Measure**
SLICER-TMU	0.7963	0.8095	0.8028
SLICER-MCT	0.7937	0.8139	0.8037
SE	0.8033	0.8673	0.8341
RaSEC	0.8358	0.8923	0.8631

TABLE II. Quantitative Comparison

V. Conclusion

In this paper, we have proposed an intelligent framework (RaSEC) for reliable and secure multi-level edge computing for industrial applications. In our proposed framework, an underlying WMSN is partitioned into small clusters. Each cluster is represented by an LOED. The LOEDs collect data from member MSNs, aggregate it to preserve the privacy of member MSNs and reduce the volumes of data, and forward to nearby LTEDs. The LTEDs are registered with nearby HLEDs through a multi-step mutual handshaking mechanism. Once the registration process is completed, the LTEDs are allowed to forward the aggregated multimedia data to the HLEDs for analysis. The HLEDs use a CNN-based architecture to analyze the aggregated data to detect the movements of objects. Simulation results have shown that our proposed framework outperforms the existing frameworks in terms of various metrics. In future, we plan to extend the scope of our proposed framework in mobile environments, e.g., mobile crowdsensing and computing. We also plan to add mathematical modeling and see the behavior of our proposed framework in large-scale networks.

References

[1] M. Usman, X. He, K.-M. Lam, M. Xu, S. M. M. Bokhari, and J. Chen, "Frame interpolation for cloud-based mobile video streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 831–839, 2016.

[2] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[3] M. Usman, X. He, K. K. Lam, M. Xu, J. Chen, S. M. M. Bokhari, and M. A. Jan, "Error concealment for cloud-based and scalable video coding of hd videos," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 975–987, 2019.

[4] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, 2018.

[5] C. Esposito, A. Castiglione, F. Pop, and K.-K. R. Choo, "Challenges of connecting edge and cloud computing: a security and forensic perspective," *IEEE Cloud Computing*, no. 2, pp. 13–17, 2017.

[6] M. Usman, X. He, M. Xu, and K. M. Lam, "Survey of error concealment techniques: Research directions and open issues," in *2015 Picture Coding Symposium (PCS)*. IEEE, 2015, pp. 233–238.

[7] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, 2017.

[8] D. He, S. Chan, and M. Guizani, "Security in the internet of things supported by mobile edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 56–61, 2018.

[9] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.

[10] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, 2018.

[11] K. Mahmood, X. Li, S. A. Chaudhry, H. Naqvi, S. Kumari, A. K. Sangaiah, and J. J. Rodrigues, "Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure," *Future Generation Computer Systems*, vol. 88, pp. 491–500, 2018.

[12] J. Abawajy, S. Huda, S. Sharmeen, M. M. Hassan, and A. Almogren, "Identifying cyber threats to mobile-iot applications in edge computing paradigm," *Future Generation Computer Systems*, vol. 89, pp. 525–538, 2018.

[13] M. Usman, M. A. Jan, X. He, and J. Chen, "A survey on representation learning efforts in cybersecurity domain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–28, 2019.

[14] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[15] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: Challenges, solutions and opportunities," *IEEE Network*, no. 99, pp. 1–8, 2018.

[16] X. Li, S. Liu, S. Kumari, and J. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted iot applications," *IEEE Internet of Things Journal*, 2018.

[17] M. Usman, M. A. Jan, X. He, and M. Alam, "Performance evaluation of high definition video streaming over mobile ad hoc networks," *Signal Processing*, vol. 148, pp. 303–313, 2018.

[18] M. Du, K. Wang, Z. Xia, and Y. Zhang, "Differential privacy preserving of training model in wireless big data with edge computing," *IEEE Transactions on Big Data*, 2018.

[19] C. Xu, J. Ren, D. Zhang, and Y. Zhang, "Distilling at the edge: A local differential privacy obfuscation framework for iot data analytics," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 20–25, 2018.

[20] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location privacy in mobile edge clouds: A chaff-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2625–2636, 2017.

[21] M. Yazici, S. Basurra, and M. Gaber, "Edge machine learning: Enabling smart internet of things applications," *Big Data and Cognitive Computing*, vol. 2, no. 3, p. 26, 2018.

[22] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, 2018.

[23] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "Ubehealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities," *IEEE Access*, vol. 6, pp. 32 258–32 285, 2018.

[24] M. Usman, M. A. Jan, A. Jolfaei, M. Xu, X. He, and J. Chen, "Daac: A distributed and anonymous data collection framework based on multi-level edge computing architecture," *IEEE Transactions on Industrial Informatics*, 2019.

[25] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 249–261, 2018.

[26] M. Usman, M. Jan, X. He, and P. Nanda, "Data sharing in secure multimedia wireless sensor networks," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2016.

[27] M. Usman, N. Yang, M. A. Jan, X. He, M. Xu, and K.-M. Lam, "A joint framework for qos and qoe for video transmission over wireless multimedia sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 746–759, 2018.

[28] M. Usman, M. A. Jan, X. He, and J. Chen, "A mobile multimedia data collection scheme for secured wireless multimedia sensor networks," *IEEE Transactions on Network Science and Engineering*, 2018.

[29] L. Li, K. Ota, M. Dong, and W. Borjigin, "Eyes in the dark: Distributed scene understanding for disaster management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3458–3471, 2017.

[30] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[31] 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.

[32] 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.