Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REINFORCING QUERY-BASED SURVEILLANCE SYNOPSIS SECURITY THROUGH DISTRIBUTED COMPUTING

HASSAN I. SAYED AHMED [1], RASHA SHOITAN [2], GHADA F. ELKABBANY [1], MONA M. MOUSSA [2], MOHAMED S. ABDALLAH [1,3, 4,*] ,YOUNG-IM CHO [3,*]

[1] Informatics Department, Electronics Research Institute (ERI), Cairo 11843, Egypt

[2] Computer and systems Department, Electronics Research Institute (ERI), Cairo 11843, Egypt

[3] Department of Computer Engineering, Gachon University, Seongnam 13415, Republic of Korea

[4] AI Lab, DeltaX Co., Ltd., 5F, 590 Gyeongin-ro, Guro-gu, Seoul 08213, Republic of Korea

E-mail: hassanibrsayed@eri.sci.eg, gelkabbany@eri.sci.eg, rasha.shoitan@eri.sci.eg, [2]mona_moussa@eri.sci.eg, yicho@gachon.ac.kr, sameer@gachon.ac.kr

ABSTRACT

Video synopsis facilitates the efficient analysis of surveillance footage by condensing extended video content into shorter segments and reorganizing moving objects temporally based on a predefined objective function. Various approaches have been developed to address challenges in video synopsis, such as object collisions, maintaining chronological order, reducing computational time, and managing scene complexity by grouping activities based on user-defined queries. However, many query-based detection techniques often overlook specific tasks like detecting intrusions, such as unauthorized entry, identifying individuals wearing unauthorized clothing, or detecting person substitution. This research extends one of the conventional query-based approaches by characterizing each object tube through its motion and visual features to enhance the detection of such intrusions. Moreover, it implements query-based video synopsis generation using a distributed architecture that utilizes multiple servers to boost performance. While the distributed model effectively mitigates and improves the query-based computational performance, it introduces potential security risks, including object tube substitution during cross-device transfers and the possibility of tube loss. The proposed system contributes by building a distributed system architecture that reduces computational complexity and provides a comprehensive solution that balances between efficiency, security, and reliability in handling surveillance video synopsis. It offers a two-layer security model; the first layer embeds a watermark within each object tube to safeguard against substitution during processing. The second layer serializes each object tube, ensuring no loss occurs during transmission, thus preserving the integrity of the entire process. This novel approach integrates distributed computing, watermarking, and serialization to enhance both the efficiency and security of the video synopsis process, providing a robust solution for large-scale surveillance video analysis. The simulation results indicate that the security module effectively extracts the watermark with a reasonable PSNR and identifies any tube loss. Additionally, the analysis reveals that utilizing a distributed system greatly enhances performance, which is critical for real-time applications. Specifically, the proposed distributed model reduces computation time by approximately 94% compared to sequential execution when implemented across twenty machines.

Keywords: *Video Synopsis; Video Surveillance; Distributed Computing; Object-Based Watermark; Tube Serialization*

1. INTRODUCTION

In recent years, the increasing demand for enhanced security in public places such as airports, banks, malls, and train stations has resulted in the global installation of numerous surveillance cameras, which continuously capture vast amounts of data 24/7. Traditionally, having one person review all video feeds from security cameras to identify specific activities is exhausting and monotonous, leading to the likelihood of missing important events. This challenge arises from the need to monitor numerous cameras concurrently. Furthermore, most security footage is uneventful, with only a few containing critical incidents. The efficient organization, searching, analysis, manipulation, and representation of this enormous volume of acquired video data necessitate

automated computational methods. In 2006, Rav-Acha et al. [1] introduced an approach called video synopsis that automatically creates short, informative summaries of long videos. Video synopsis aims to intelligently condense the original video while preserving all the essential video activities. The condensed video is produced by simultaneously displaying all objects despite their occurrence at distinct time points. Video synopsis differs from standard video summarization approaches in that it creates a coherent narrative that captures the core of the video's content rather than simply collecting important frames or clips from it. It entails reducing the entire video into a significantly shorter version while retaining the storyline, key events, and main ideas. Video synopsis strives to provide viewers with a thorough comprehension of the video's content in a fraction of the time it takes to watch the entire video, making it a valuable tool for quickly capturing the fundamental essence of longer videos. Video synopsis frameworks follow different steps to summarize lengthy video footage efficiently. First, a timelapse of the static background is created. Next, intelligent algorithms are employed to determine the optimal way to condense essential events that occurred at different times into a shorter clip. This process involves tracking and isolating moving objects from the background throughout the video before finally stitching everything together seamlessly for a concise and informative summary [2]–[4].

Video synopsis techniques possess several measurable characteristics that guarantee their efficacy. The primary objective is to capture the most pertinent activity while avoiding unnecessary repetition. Furthermore, it is crucial to maintain the chronological order and spatial relationships of objects in the condensed video. Moreover, it is essential to ensure minimal overlap between moving objects to prevent potential confusion. Lastly, the summary should be flawless so readers can easily follow along without being sidetracked. Recent research in video synopsis techniques has produced a variety of approaches, each addressing specific challenges in compressing lengthy videos. Some researchers propose an energy minimization function and optimization algorithms for minimizing overlaps between moving objects and preserving objects' chronological order and spatial relationships in the condensed video [5]–[14].

Further, some approaches tackle the detection and tracking failures that impact the video synopsis performance [15]–[17]. Additionally, researchers attempt to make video synopsis generation faster and more accessible by utilizing parallel processing, which is crucial for real-time applications and large-scale video analysis using distributed computing. To the best of the author's knowledge, a limited number of research endeavors are dedicated to expediting the computational speed of video synopsis. Lin et al. [18] developed a distributed processing method to reduce the computation time for video synopsis. Although this approach utilized the multi-processing system's computing power and multi-thread capabilities to reduce the video synopsis processing time, it did not address the substitution or loss of object tubes during the transmission process.

Other approaches seek to improve the visual quality of the synopsis by addressing scene complexity by grouping similar activities based on movement direction, action type, and clothing colours. Ahmed et al. [19] developed a video synopsis technique for traffic monitoring using a user query based on object attributes. Initially, moving objects are tracked and classified into categories like cars, pedestrians, and bikes using deep learning. The user then provides a query, and the corresponding object tubes are merged onto a background frame to generate the synopsis. Additionally, Namitha et al. [20] introduced an interactive visualization technique for synopsis video generation, utilizing basic visual features like colour, size and spatial features to retrieve specific objects. YOLOv3 and Deep-SORT are employed for detection and tracking, with tube grouping preserving object relationships and a space-time cube algorithm arranging the tube groups within a predefined synopsis length. Pritch et al. [21] proposed a real-time video synopsis based on user queries to display activities within a specific timeframe from continuous surveillance or webcam footage. In a separate study, Pritch et al. [22] presented a video synopsis technique highlighting comparable activities using consistent appearance and motion features. While these clustering and user query-based methods effectively address the challenge of creating satisfactory synopsis videos in crowded scenes by mitigating collisions, they overlook specific appearance attributes such as gender, age, carrying items, having a baby buggy, and clothing colours. Thus, Shoitan et al. [23] provide a technique for creating a video synopsis based on user-defined queries. This approach incorporates motion descriptions and particular visual appearance parameters, like gender, age, carrying objects, having a baby carriage, and upper and lower clothing colour. YOLOX and ByteTrack are employed for efficient person detection and

tracking, creating object tubes [24], [25]. After gathering the appropriate persons, a whale optimization method is utilized to organize them chronologically, reducing the chances of collisions and guaranteeing a brief summary video.

Most of the methods discussed focus on general query-based detection techniques and do not specifically address the retrieval or detection of intrusions that have already occurred in a video, such as detecting when a person or object entered a restricted area without proper authorization, identifying someone wearing unauthorized clothing, or recognizing instances of person substitution which is a weak point in these researches. To address this problem, the proposed research is designed based on the method mentioned [23], which describes each tube by its motion and visual description for detecting the intrusion. In addition, this proposed research offers a solution for the great processing time needed to generate a synopsis video. It implements a query-based video synopsis by utilizing multiple servers to harness its computing capabilities fully. While the proposed distributed system effectively mitigates the computational complexity involved in video synopsis generation, it introduces the risk of object tube substitution during the distribution of tubes across different devices and the potential for tube loss. To address these challenges, the proposed method introduces a two-layer security model. The first layer involves embedding a watermark within each tube to prevent substitution during processing. The second layer serializes the object tubes to detect any transmission loss, ensuring their integrity throughout the process. This novel approach integrates distributed computing, watermarking, and serialization to enhance the efficiency and reliability of the video synopsis process, presenting a promising solution for large-scale surveillance video analysis. The primary contributions of this work are as follows:

1. Presenting a comprehensive solution that balances efficiency, security, and reliability in handling surveillance video synopsis across multiple servers, addressing computational and integrity concerns.
2. Proposing a distributed system architecture that reduces the computational complexity in generating query-based video synopses to enhance the efficiency of large-scale video surveillance analysis.
3. Introducing the two-layer security model:
   - Watermarking for tube integrity: A watermark is embedded within each video tube to prevent unauthorized tube substitution during distributed processing.
   - Tube serialization for loss detection: Tube serialization is implemented to track and detect any loss of object tubes during transmission between devices.
4. Concentrating on detecting intrusions in query-based video synopsis, such as unauthorized entry into restricted areas, wearing unauthorized clothing, and person substitution, utilizing motion and visual descriptions of object tubes.

The paper's structure is as follows: Section 2 presents the details of the proposed secure parallel computing framework. Then, in section 3, the experimental results and their analysis are discussed. Finally, in section 4, the conclusions and future research directions are found.

2. THE PROPOSED FRAMWORK

Scene complexity challenges video synopsis performance by making it difficult to accurately track and prioritize multiple overlapping objects, leading to potential occlusions and loss of clarity. Diverse object behaviors, dynamic backgrounds, and frequent scene changes further complicate the process, making it hard to maintain temporal and spatial coherence. These factors increase computational demands, often making video summaries less effective and confusing. One of the solutions for the scene complexity challenge is the query-based video synopsis, which generates the synopsis based on a specific attribute. All the conventional query-based methods retrieve the persons without concentrating on intrusion detection, including unauthorized entry into restricted areas, identifying individuals wearing unauthorized clothing, or recognizing instances of person substitution.

Therefore, this work focuses on a query-based approach for intrusion detection when dealing with lengthy surveillance videos. To enhance the performance of the query-based method the proposed solution is implemented using a distributed system. Expanding on the query-based video synopsis method outlined in [23], the proposed approach modifies it to focus on detecting individuals wearing unauthorized clothing or identifying cases of person substitution.

Moreover, it presents a distributed query-based video synopsis model that uses multiple nodes to enhance processing power and improve performance. However, parallelizing the synopsis process across multiple nodes increases the risk of object tube substitution or loss. To mitigate these issues, the proposed method presents two layers of

security. First, to ensure the authenticity of individuals in the video and prevent any unauthorized substitution, watermarks are embedded within each object tube. Then, a serialization technique is introduced to monitor and manage the presence of object tubes during the parallelization procedure to minimize the risk of object tube loss. By combining a distributed system with object tube watermarking and serialization, the proposed approach addresses the challenges of computational complexity as well as the risks of object tube substitution or loss in the query-based video synopsis process. This approach aims to improve efficiency, reliability, and security of the query-based video synopsis generation in large-scale surveillance applications. Figure 1 presents the proposed secure distributed architecture for query-based video synopsis.

2.1 The Proposed Security Model

Person tubes in video synopsis contain critical information that must be securely protected against theft or tampering. Due to the parallelized processing, there is a risk of object tubes being lost or substituted with those of other individuals, which could deceive law enforcement. To address this issue, the paper proposes a two-layer security model. The first layer involves watermark embedding, where a watermark is embedded into every frame of each "tube". This ensures that any unauthorized modifications or insertions can be detected. The second layer is tube serialization, where a unique serial number is assigned to each tube in the video to enable individual tracking and authentication of each tube. The following subsection will describe the details of this proposed security model and its layers.

2.1.1 First security layer: watermarking

In this layer, a watermark is embedded within each frame of every tube to safeguard them from substitution or any attacks. This study implements the Quantization Index Modulation (QIM) watermarking technique on the object tubes to ensure protection. The QIM watermarking scheme offers several key advantages, making it highly effective for secure media watermarking. QIM is known for its low computational complexity, making it suitable for real-time applications with minimal processing power. Additionally, it supports a high embedding capacity, allowing more information to be hidden without affecting the host content. It offers enhanced security, making the watermark difficult to detect or alter without knowledge of the quantization parameters. These features make QIM ideal for digital watermarking in various media formats. The following steps provide a detailed explanation of how QIM watermark embedding and extraction are performed for each tube [26].

- **The Watermarking Embedding Process**

The watermark embedding process based on QIM modify the pixel values of a host image based on a quantization step size and the watermark data. The goal is to embed the watermark imperceptibly to human eyes but can be extracted later [27]. As illustrated in Figure 2, the embedding process can be described through the following steps:

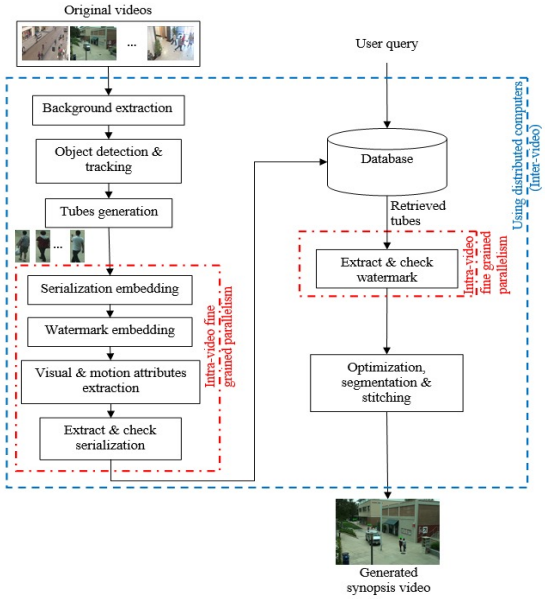


Figure 1: The proposed Secure Distributed Architecture for Query-Based Video Synopsis

Step 1: Prepare the host (frame) and the watermark

In the proposed method, the watermark is embedded in the blue channel (host signal) of the RGB digital host image. The blue channel is chosen for its ability to maintain imperceptibility, ensuring the watermark does not affect the visual quality. Let I(i,j) represents the pixel value at position (i,j) in the blue channel of the image. The watermark is a binary image, where each pixel value "w(i,j)" is either "0 or 1". This binary value dictates how the corresponding pixel in the host signal is altered. The watermark image is resized to match the host image dimensions, streamlining the extraction process.

Step 2: Choose a Quantization Step Size (Δ)

The quantization step size Δ is crucial as it determines the granularity of quantization. A

larger Δ provides higher robustness but might make the watermark more noticeable. Conversely, a smaller Δ results in less visible changes to the image but might reduce the robustness of the watermark. In this research, a small quantization step size is chosen to make the watermark fragile, allowing it to detect any alterations made to the tube [28], [29].

Step 3: Quantize the Pixel Values

Quantization involves mapping the original pixel values to the nearest multiple of Δ as follows:

$$Q(I_w(i,j),\Delta) = I_w(i,j) - (I_w(i,j) \bmod \Delta) \quad (1)$$

Where Q($I(i,j)$, Δ) is the quantized value of the pixel $I(i,j)$. The quantization function is key to the QIM technique, as it controls how much the tube frame is modified [27].

Step 4: Embed the Watermark

The watermark is embedded by adjusting the quantized pixel values based on the watermark bit $w(i,j)$.

If $w(i,j)$=0: $$Iw(i,j) = Q(I(i,j),\Delta) \quad (2)$$

The pixel value is mapped directly to the nearest multiple of Δ

If $w(i,j)$=1: $$Iw(i,j) = Q(I(i,j),\Delta) + 2\Delta \quad (3)$$

The pixel value is shifted by $\frac{\Delta}{2}$ to embed the watermark bit.

The embedding equation is a combination of equation 2 & 3

$$Iw(i,j) = Q(I(i,j),\Delta) + w(i,j)\cdot\frac{\Delta}{2} \quad (4)$$

In (4) the pixel value is modulated based on the watermark bit.

Step 5: Clipping the Watermarked Pixel Values

After embedding, it is important to ensure that the modified pixel values remain within the valid range of [0,255] for 8-bit images. This is done using a clipping operation as follows:

$$Wreconstructed(i,j) = w(i,j) \times 255 \quad (5)$$

Clipping is essential to prevent any overflow or underflow in pixel values, which could lead to visible artifacts in the watermarked image. The QIM method is designed to minimize such artifacts while maintaining the integrity of the watermark.

Step 6: Replace the Modified Channel

After embedding the watermark into the blue channel, this modified blue channel is merged back with the other original green and red channels to form the final watermarked image as follows:

$$Watermarked\ Image = merge(I_w, G, R) \quad (6)$$

I_w is the watermarked blue channel, and G and R are the unmodified green and red channels.

- **The Watermarking Extraction and Detection Process**

Watermark extraction in QIM entails a sequence of steps to retrieve the watermark from the watermarked tubes, reversing the embedding process to recover the embedded bits. If the tube has been modified, the extraction process can expose inconsistencies, signaling possible tampering or substitution. The detailed steps for extracting the watermark using QIM, as outlined in figure 3, are as follows:

Step 1: Quantization of Watermarked Signal

The initial step in the extraction process is to quantize the pixel values of the watermarked image in the same manner as in the embedding stage. This step identifies the nearest quantization level for each pixel. The quantization equation is:

$$Q(I(i,j),\Delta) = I(i,j) - (I(i,j) \bmod \Delta) \quad (7)$$

Step 2: Determine the Watermark Bit

After quantization, the watermark bit is determined based on the difference between the watermarked pixel value and the quantized value. The extraction rules: if the remainder of the watermarked pixel value $\mathbf{I_w(i,j)}$ modulo Δ is less than $\frac{\Delta}{2}$:

$$Extracted\ w(i,j) = 0 \quad (8)$$

If the remainder of the watermarked pixel value $I_w(i,j)$ modulo Δ is less than $\frac{\Delta}{2}$:

$$Extracted\ w(i,j) = 1 \quad (9)$$

Combined Extraction Equation:

$$w(i,j) = \begin{cases} 0 & if\ (I_w(i,j)\ \ mod\ \ \Delta) < \frac{\Delta}{2} \\ 1 & if\ (I_w(i,j)\ \ mod\ \ \Delta) \geq \frac{\Delta}{2} \end{cases} \quad (10)$$

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

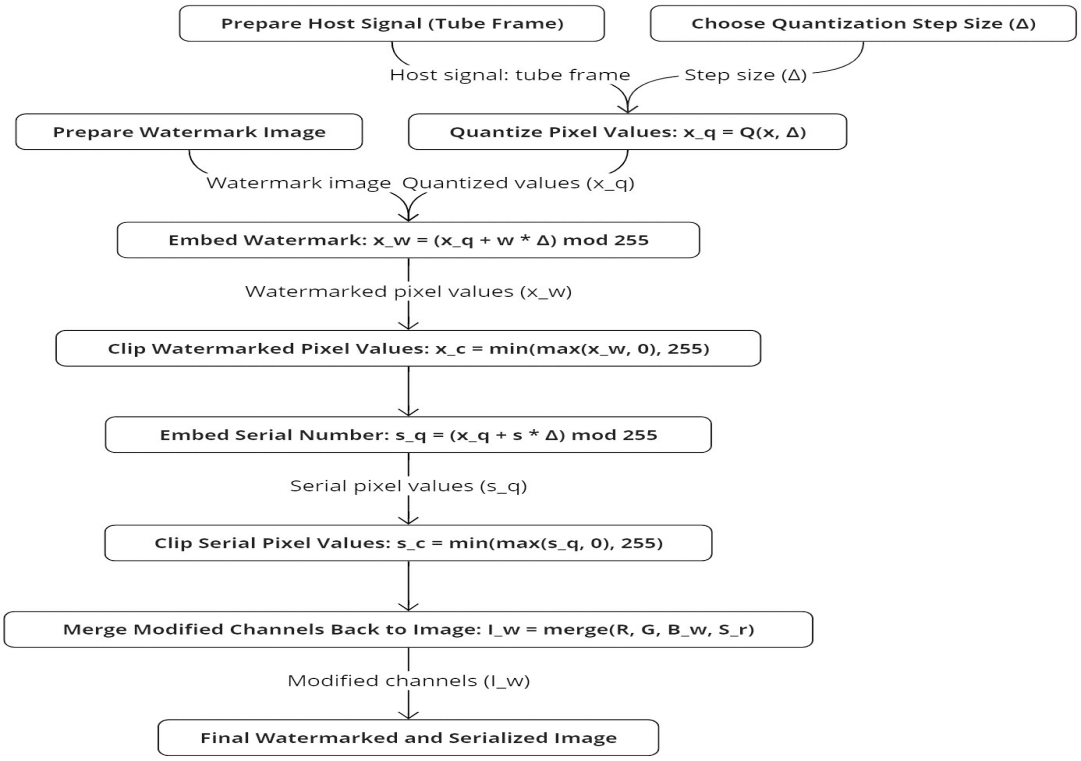


Figure 2: The proposed watermark and serialization embedding process.

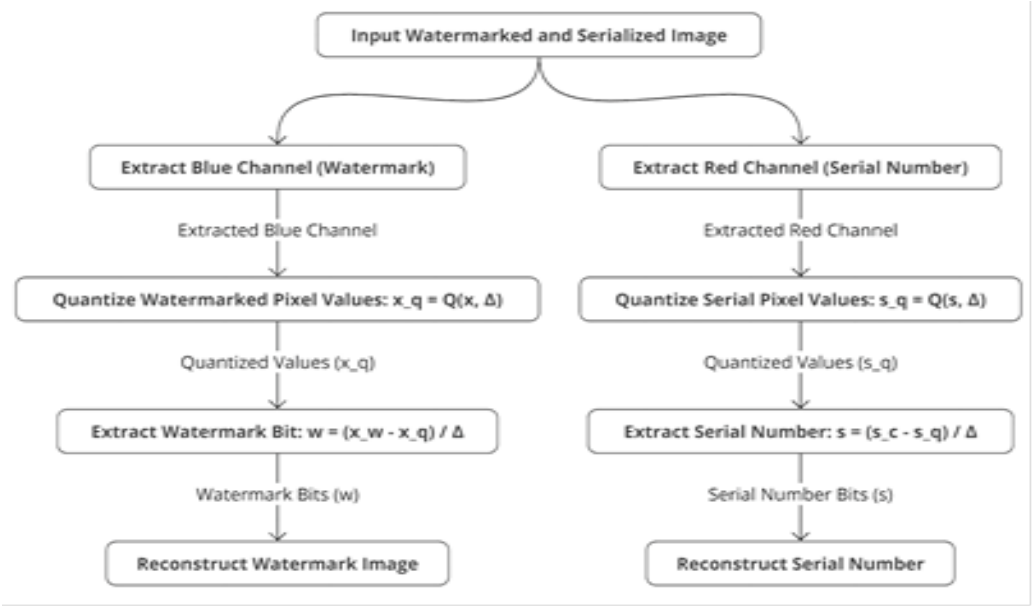


Figure 3 The proposed watermark and serialization extraction process

The combined extraction equation in QIM is an essential component of the watermark extraction process. It operates by analyzing the remainder when the watermarked pixel value "$w(i,j)$" is divided by the quantization step size Δ. This remainder indicates how the original pixel was modified during the watermark embedding phase. The equation determines the embedded watermark bit by comparing the remainder to a threshold value of $\frac{\Delta}{2}$. If the remainder is less than this threshold, the watermark bit is interpreted as 0; else if it is greater than or equal to $\frac{\Delta}{2}$, the bit is interpreted as 1. This approach is simple and efficient, making it ideal for video synopsis applications, as it involves minimal computation.

Step 3: Reconstruction of the Watermark

Once all the bits are extracted from the watermarked image, they are used to reconstruct the watermark. The watermark image is recreated by arranging the extracted bits in the same format as the original watermark. Equation 11 represents the reconstructed watermark.

$$Wreconstructed(i,j) = w(i,j) \times 255 \quad (11)$$

$Wreconstructed(i,j)$ is the pixel value at position(i,j) in the reconstructed watermark image.

2.1.2 The second security layer: Tube serialization

Dropping attacks involve the removal or loss of tubes, either intentionally by an attacker or due to communication errors through parallel processing or the distributed processing model proposed in this research. To address the issue of tube loss, an additional layer of security through serialization is incorporated into the proposed method. The serialization process involves assigning a unique serial number to each tube and embedding this number into the digital representation of the tube. This enhances traceability, integrity, and security, ensuring that each tube can be tracked and authenticated individually. This process begins by assigning each tube a unique serial number, ensuring that each item can be distinctly recognized. For example, if there are 100 tubes, they would be sequentially numbered from 1 to 100. The next critical step involves embedding these serial numbers into the digital representation of the tubes. This embedding can be achieved using the QIM technique, similar to tube watermarking, where the serial number is encoded into the tube in a way that is not visually noticeable but can be reliably extracted when required. This approach allows for easy identification and tracking of each tube throughout its lifecycle. In a scenario where tubes are serialized and transmitted over a network using QIM, the process involves the following steps:

* Assign a serial number S to a tube.
* Use QIM to embed S into the red channel of each frame I of the tube.
* Transmit the serialized image I over the distributed processing framework.
* At the receiving end, extract the serial number S' from the received image by reading the quantized pixel values from the red channel.
* Compare S' with the expected serial number. Any mismatch indicates potential tampering or transmission error.

This method enhances the overall security and reliability of the tube integration process, ensuring that any loss or alteration of tubes can be detected and addressed. The detailed procedure for embedding and extracting the serial number is outlined in the following subsection:

- **Embedding Serial Number**

In this layer, QIM is used to embed a serial number into an image's red channel. QIM can modulate the pixel values such that the serial number can be extracted later. Embedding the serial number into the digital representation consists of the following steps, as illustrated in figure 2.

Step 1: Preliminary setup for QIM-based serialization

In video synopsis production, a serial number is generated for each tube. The process begins by selecting this serial number S' represented as an integer. This serial number is converted into an 8-bit binary string; for instance, if "S=100', its binary representation would be "S_{bin}= 01100100'. The red channel of the image, denoted as "R', is chosen as the location where the serial number is embedded. The embedding process involves adjusting the pixel values in this red channel according to a quantization step size Δ. Through this process, each tube can be uniquely identified and tracked, facilitating precise management and organization of the content.

Step 2: Embedding serial number process using QIM

The embedding process involves modulating the red channel's pixel values based on the serial number's binary representation. Each bit of the binary serial number is embedded into the corresponding pixel by quantizing the pixel value; the quantization function Q is defined as:

$$Q(x,b) = (\frac{x}{\Delta} \times \Delta) + \frac{\Delta}{2} \times b \quad (12)$$

Where "x" is the original pixel value, Δ is the quantization step size, "b" is the bit to be embedded (either 0 or 1). This function ensures that: if "b=0", the pixel value 'x" is quantized to the lower quantization level. While, if "b=1" the pixel value "x" is quantized to the upper quantization level. In the context of embedding each bit of the serial number, for each bit "b" of the serial number's binary representation, the corresponding pixel value in the red channel is adjusted using the QIM quantization function. If "$S_{bin}=b_1,b_2,\ldots,b_8$" and the first 8 pixels of the red channel are denoted as "$R(1,1), R(1,2), \ldots, R(1,8)$", the embedding process is "$R_{new}(1,j) = Q(R(1,j),b_j)$". Where "$j$=1, 2, …,8", and each "$R_{new}(1,j)$" is the new pixel value after embedding the bit "bj" .

- **Serial Number Extraction Process**

The extraction process involves reading the quantized pixel values from the red channel and reconstructing the binary serial number as illustrated in figure 3.

For each pixel "$R_{new}(1,j)$" in the red channel, the embedded bit "b_j" is extracted by comparing the pixel value to the quantization step Δ:

$$b_j = \begin{cases} 0 & if\ R_{new}(1,j)\ \%\ \Delta < \frac{\Delta}{2} \\ 1 & if\ R_{new}(1,j)\ \%\ \Delta \geq \frac{\Delta}{2} \end{cases} \quad (13)$$

In the context of reconstructing the serial number, the extracted bits "$b_1,b_2,\ldots,b_8$" are then combined to reconstruct the serial number based on (13):

$$S_{extracted} = b_1 \times 2^7 + b_2 \times 2^6 + \ldots + b_8 \times 2^0 \quad (14)$$

From (14), this binary number is then converted back into its decimal form to retrieve the original serial number.

2.2 The Parallel Secure Video Synopsis Model

Parallelization is categorized as: task-based Parallelism (TP), and Data-based Parallelism (DP). In TP, the sequential application/program/ instructions decomposed into small parts that could be computed in parallel across multiple nodes/processors (using the same dataflow). On the other hand, in TD, the input data is divided into smaller independent subsets that could be performed concurrently [30]–[32]. Various video processes need complex computations which makes them suitable for parallelization. Researchers exploit the benefits of parallel computing to efficiently execute video algorithms such as encoders [33], video compression [34], video transcoding [35], and video synopsis [18]. Moreover, distributed cloud computing was used speed up person re-identification task [36]. These studies have taken diverse approaches regarding parallel architecture and the specific types of parallelism employed.

In this work, a two-tiered parallel model is proposed to enhance the overall performance of the proposed secure version of the selected video synopsis system [23]. This model effectively leverages parallelism at both the inter-video (coarse-grained parallelism) and intra-video (fine-grained parallelism) levels to maximize the efficiency of the selected algorithm. At the coarse-grained level, all videos are sorted according to their execution time in descending order and concurrently executed across a multi-node model. Then, at the fine-grained level, multiple machines within each node cooperate to execute individual videos collectively. Several key factors are considered to achieve parallel computing advantages, including the optimal video allocation between nodes, the precedence constraints between different tasks inside each video, and the communication overhead due to task distribution inside each node. The parallel implementation is assumed to run on a computer model equipped with "N" homogenous nodes. Inside each node, there are "M" homogeneous machines. Moreover, the proposed model is undertaking "V" videos of varying sizes and durations. Each video goes through ten processing steps, namely: background extraction, person detection and tracking, person tube generation, serialization embedding, watermark embedding, visual and motion feature extraction, serialization extraction, query-based object tube retrieval, watermark extraction, and finally, optimization, segmentation, and stitching. Based on the total time required to execute these tasks, these videos are assigned priorities and arranged in descending order, with the video needing the longest execution time having the highest priority. The first three steps must be executed sequentially, while step six, which is the

most compute-intensive task (needs a long time to be performed), can be executed concurrently. On the other hand, the fourth and fifth steps involve adding watermarks and serial numbers to each frame within each object tube. The execution time required for these steps varies according to the number of tubes and frames, so they can be executed in parallel if there is a large number of a frame or segments. Otherwise, they are executed sequentially. If these two steps are achieved, then they are carried out on parallel lines within the video, and if they are not achieved, they are carried out sequentially. While step 7 must be executed sequentially, steps 8 and 9 can be executed in parallel if a certain threshold is met; otherwise, they are executed sequentially. Finally, step 10 must be done sequentially. For each video "v_i" for steps 4,5,6,8, and 9, each one of them can be decomposed into subtasks that are capable of parallel execution. Figure 4 illustrates the steps of the proposed distributed secure video synopsis model, which must be applied to each video (v_i where $1 \le i \le V$).

Initially, all videos are sorted based on their execution times in a descending manner and placed into a scheduling list. Since this work is conducted offline, the 'V' videos are distributed across different nodes to achieve balanced workloads. Then, each node adds its assigned videos to its local queue (coarse-grained). Parallelization is performed inside each node to optimize the performance within a given architecture, where the 'M' machines collaborate to execute their respective assigned videos (fine-grained).

As part of the parallel distribution process, some video segments (tubes) may need to be transferred between different machines within the same node. This data transfer introduces a risk of tube loss. The proposed model uses tube serialization to avoid tube loss due to data transfer or wipeout by external hackers. That is to say, one of the main issues of the proposed technique is protecting the transferred data from loss or elimination during the parallel distribution workflow. The serialization helps ensure the integrity and security of the video content as it is processed across the distributed node infrastructure. As shown in figure 4, a multi-node system containing "*N*" nodes is assumed, where each node contains '*M*" machines, and assuming that the number of videos is "*V*". There are three cases:

(i) $N > V$: one video is assigned to each node from "$Node_1$" to "$Node_V$" and the other "*N-V*"" nodes will be idle.

(ii) $N = V$: one video is assigned to each node.

(iii) $N < V$: since this work is conducted offline, the "*V*" videos are distributed across different nodes to achieve balanced workloads.

Assuming T_s is the total sequential time (hours), T_{comp} is computation time (hours), T_{ov} is overhead time (sec) and T_{par} is parallel execution time (hours) The total sequential time

$$T_s = \sum_{i=1}^{V} T_{vi} \quad (15)$$

To calculate the computation time for one video "T_{vi}", the following equation is used:

$$T_{vi} = \sum_{i=1}^{10} T_{i10} \quad (16)$$

Where: "T_{i1}", "T_{i2}", and "T_{i3}" are the time to calculate background extraction, person detection and tracking and person tube generation, "T_{i4}" is the time to embed a serial number into different object tubes and "T_{i5}" is the time needed to embed watermarks in a different frame of the object tubes, "T_{i6}" is the time needed to execute visual and motion feature extraction. "T_{i7}" is the time needed to extract the serial number. "T_{i8}" is the time to execute query-based object tube retrieval. "T_{i9}" is the time to extract the watermark. Finally, "T_{i10}" is the time to execute optimization, segmentation and stitching. To calculate "T_{par}", the following equation is used:

$$T_{par} = T_{comp} + T_{ov} \quad (17)$$

Where

$$T_{comp} = \max \{T_{comp}(node_j)\} \quad 1 \le j < N \quad (18)$$

and

T_{ov} can be computed using the following equation:

$$T_{ov} = T_c + T_g + T_{app} \quad (19)$$

Where: "T_c" is the local communication overhead, the time to access memory (memory contention and synchronization). "T_g" is the global communication overhead inside each node, the time spent on inter-communication. "T_{app}" is the application overhead time, the wasted time due to application dependency [37], [38]. As mentioned above, inside each node, more than one machine cooperates to execute internal videos. These machines share the same memory, decreasing the communication overhead. Since "T_{i4}, T_{i5}, T_{i6}, T_{i8}", and "T_{i9}" tasks can be divided into parts and calculated in parallel, for video "v_i," more than one machine executes these tasks for different tubes.

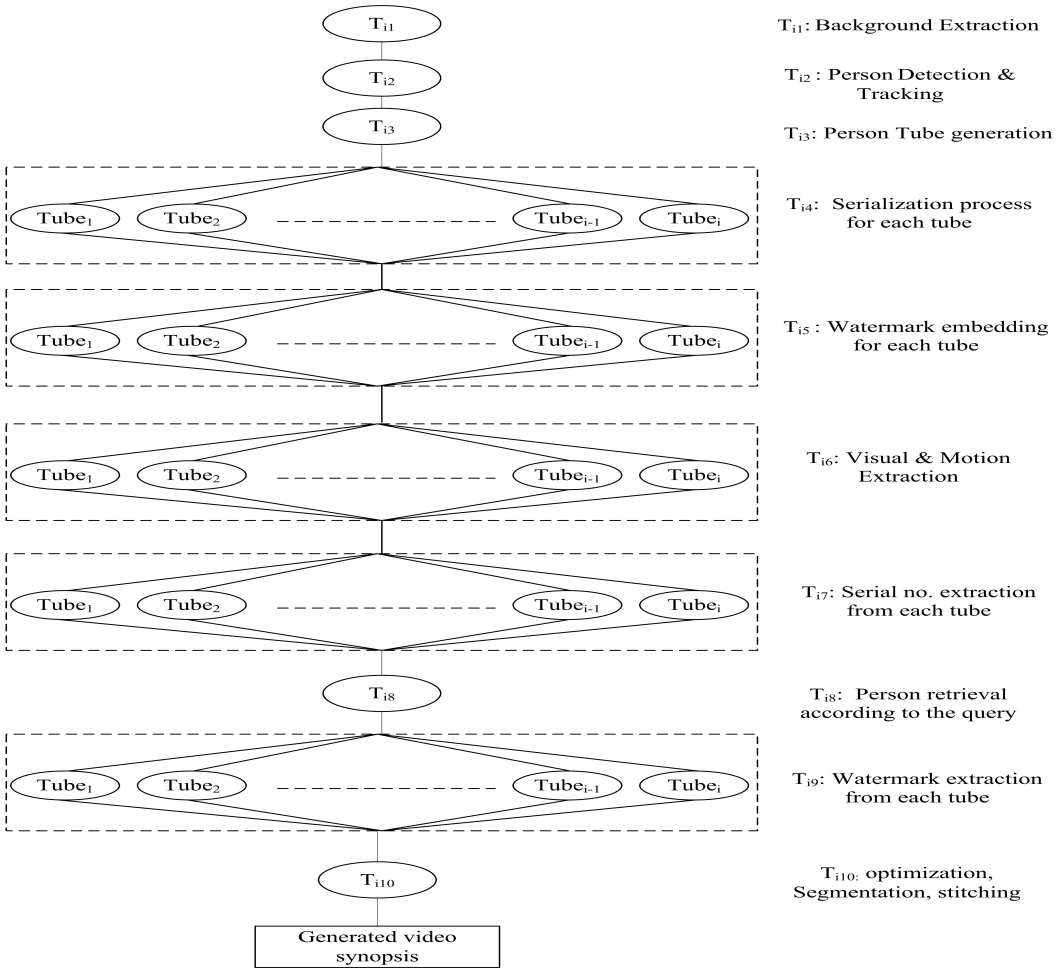


Figure 4: The Steps of The Proposed Distributed Secure Video Synopsis Model (These Steps Must Be Done For Each Video 'Vi', Where 1≤I≤V)

(a)

(b)

(c)

(d)

Figure 5: Tube Frames Samples & Watermark (a) MCT Datasets, (b) VIRAT, (c) Oxford Town Center, (d)Watermark

Inside $Node_j$, assume that there are "h" videos. Each video "v_{hj}" has "L_{hj}" tubes and "h" machines cooperate to execute this video. Each machine computes $\left|\frac{L_{hj}}{h}\right|$ tubes, and the remaining $\left\{L_{hj} - \left[\left|\frac{L_{hj}}{h}\right| * h\right]\right\}$ tubes are assigned to the first $\left\{L_{hj} - \left[\left|\frac{L_{hj}}{h}\right| * h\right]\right\}$ machines.

The next section studies and evaluates the experimental results of the proposed secure parallel video synopsis framework.

3. EXPERIMENTAL RESULTS

In this work, a set of experiments using diverse surveillance video datasets is done to evaluate the performance of the proposed secure parallel video synopsis framework. Additionally, an analysis of the experimental results is presented and discussed.

3.1 Experimental Setup

3.1.1 Dataset

To evaluate the proposed system, three publicly available datasets are utilized: Oxford Town Center dataset [39], Multi-Camera Object Tracking (MCT) dataset [20] and Video and Image Retrieval and Analysis Tool (VIRAT) dataset [40]. The first comprises CCTV footage captured at a busy Oxford intersection, featuring pedestrians and recorded at 25 frames per second with a resolution of 1920x1080 pixels and the video encompasses 7502 frames. The second dataset is composed of four subsets, each containing multiple cameras with varying resolutions and environmental conditions (indoor and outdoor). For this research, video sequences from MCT dataset are selected, featuring 234 individuals across 24,000 frames captured for 20 minutes. The third dataset is notable for its realism, capturing natural scenes with minimal actor-directed actions, and featuring cluttered backgrounds, incidental movers, and background activities. It offers diverse data collected from various sites across the USA, encompassing multiple camera viewpoints and resolutions, with actions performed by a wide range of individuals. Samples from those datasets are presented in figure 5.

3.1.2 Results analysis and discussion

This subsection provides a quantitative analysis of the proposed security and distributed models, drawing upon the numerical values derived from the evaluation metrics.

- **The proposed security model**

For measuring the performance of the proposed security model, the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Bit Error Rate (BER), Normalized Correlation (NC) metrics are applied to the watermarked-serialized tubes. The low values of PSNR, SSIM, and NC, along with high values of BER, indicate that the watermarked-serialized tubes have been impacted either by the watermark embedding process or by attacks. PSNR and SSIM metrics are shown in figure 6 (a), and (b). These figures show that the watermark embedding process generally maintains the quality and structural integrity of the watermarked serialized tubes. The PSNR values fluctuated, but generally stay above 25 dB, with many instances reaching up to 50 dB, indicating that the watermarked images closely resemble the original images. The SSIM values, which mostly stay close to 1, reinforce this observation by showing that the structural similarity between the original and watermarked tubes is largely preserved. However, the dips in both PSNR and SSIM at certain image indices suggest that while the process is effective overall, there are instances where the watermarks introduce noticeable changes to the watermarked tubes quality.

Furthermore, the BER and NC metrics provide further details on the accuracy of the watermarked serialized embedding process. The BER values, which generally hover around 0.26 to 0.28, indicate a relatively stable error rate, though there are spikes in certain frames that point to higher bit errors.

Despite these fluctuations, the overall consistency in BER suggests that while some errors are presented, the embedding process is reasonably accurate. The NC values, which vary more significantly, highlight that the correlation between the original and the watermarked serialized tubes is strong for some images but weaker for others. The spikes in NC indicate successful embedding in certain cases, but the lower values for other images suggest that the embedding process may need further refinement to ensure consistently high accuracy. These results conclude that both the watermark and serial number embedding processes are largely successful, with the serial number embedding showing perfect accuracy across all tubes. While the watermark embedding is generally effective, the variability in PSNR, SSIM, BER, and NC indicates that there is room for improvement,

particularly in ensuring consistent performance across all tubes.

Moreover, in figure 6 (e), the serial number accuracy representation is notably significant. The flat line at 1.0 across all tubes indicates that the serial number is perfectly embedded and extracted for every single frame in the dataset. This 100% accuracy rate demonstrates the robustness and reliability of the serial number embedding process, which is critical in applications where precise identification and tracking of images are necessary.

It can be noticed from figure 7 (a), the PSNR values presented in the figures indicate that a significant portion of the frames analyzed have PSNR values above 20 dB in the case of no attack. This suggests that, for the majority of the tube frames, the extracted watermark is relatively close to the original watermark in terms of quality. Higher PSNR values generally imply that the watermark has been extracted with minimal distortion, preserving the fidelity of the original watermark. In this analysis, we observe that most of the PSNR values are concentrated above the 20 dB, indicating successful extraction where the extracted watermark closely resembles the original. Although there are some fluctuations with lower PSNR values, which suggest that a few frames experienced more substantial quality loss during extraction, the overall trend shows that the watermark extraction process is effective for the majority of the images. This consistency in achieving higher PSNR values confirms that the tube frame watermarks extracted are generally of good quality, making them reliable representations of the original watermarks embedded in the frames. In addition to the PSNR analysis, the SSIM values further support the conclusion that the tube frame watermarks extracted are close to the original watermarks. It can be shown from figure7 (b) that the SSIM values for the majority of the tube frames are close to 0.9 or higher. This indicates that the structural integrity of the watermarks has been largely preserved during the extraction process. The occasional dips in SSIM values; where the index falls below 0.9; may indicate specific frames; where the structural similarity is compromised to some degree. However, the fact that the majority of the frames exhibit high SSIM values indicates that the extraction process is generally effective in maintaining the key characteristics of the original watermarks across the tube frames. These results are further validated by the NC metric and BER in figure 7 (c), (d).

To assess the impact of various attacks on the watermark, different attacks are applied to the watermarked images. The watermark is then extracted, and relevant metrics are measured for the extracted watermark. The results of these evaluations are presented in table 1. This table illustrates that PSNR, SSIM, NC, and BER show substantial differences between the original image and the extracted watermark. In the absence of any attack, high PSNR, SSIM, and NC values indicate that the watermark quality and integrity are well-preserved, while a low BER suggests minimal extraction errors. However, attacks such as noise, blurring, cropping, and scaling cause a sharp drop in PSNR and SSIM, lower NC, and higher BER, indicating substantial watermark degradation and loss of accuracy. Each attack leaves a distinct pattern, making it easy to identify the type of attack based on the changes in these metrics. Therefore, when any manipulation is made to the watermark, it becomes distorted, and the alteration can be detected, indicating that the watermark has been substituted or attacked.

Table 1: Impact of Various Attacks on the Watermark.

Attack Type	PSNR	SSIM	NC	BER
No Attack	21.91	0.90	1.00	0.23
Noisy	3.21	0.01	0.59	0.50
Blurred	4.38	0.06	0.71	0.45
Cropped	3.24	0.01	0.60	0.49
Scaled	4.38	0.05	0.71	0.45
Compressed	3.29	0.01	0.62	0.51

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

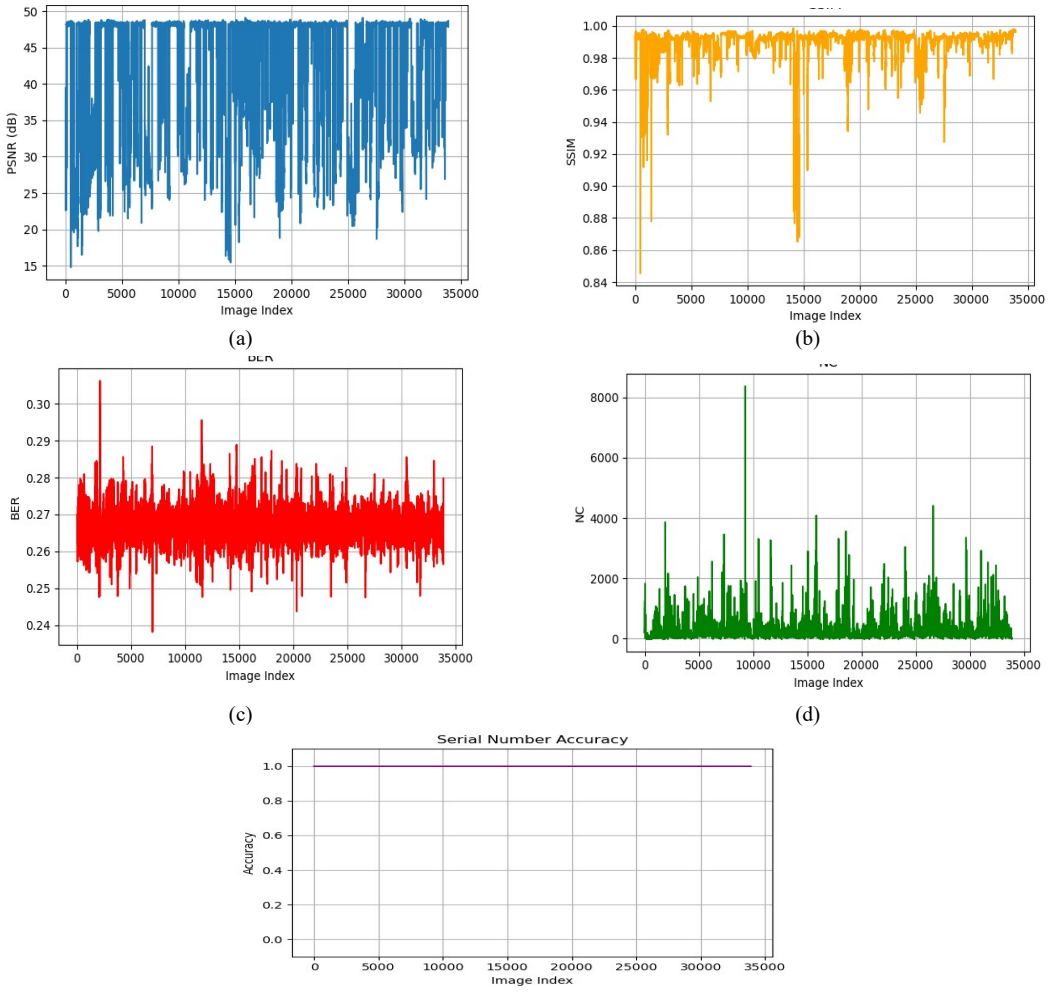


Figure 6: Original and Watermarked Serialized Tube Frame Evaluation Metrics (a) PSNR, (b) SSIM, (c) BER, (d) NC, and (e) Serialization Embedding and Extraction Accuracy

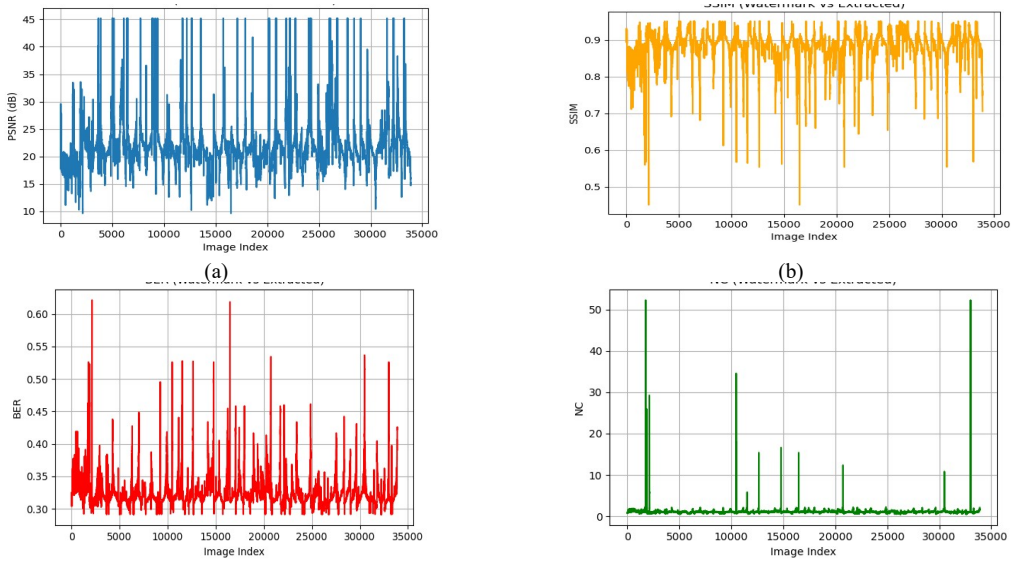


Figure 7. Watermark Extraction Evaluation Metrics (a) PSNR, (b) SSIM, (c) BER, and (d) NC

- ***The proposed distributed model***

This work employs an experimental modeling approach to implement the proposed parallel synopsis system. As discussed in Section 3.2 the parallel model operates on a node of "M' machines, each equipped with four high-speed CPUs and an optimized CPU-to-memory configuration, making it well-suited for video processing applications. Various performance metrics are employed to assess parallel systems, including execution time, speedup, efficiency, communication overhead, scalability, and improvement degree [31], [32], [41], [42].

- Execution time (parallel time) 'T_{par}' refers to the total time of the program execution.
- Speedup 'S_P' quantifies how much faster the parallel execution is compared to the sequential one. It is the ratio of the time taken by a certain task on a single node 'T_S' to the time taken by the same task on 'M' nodes 'T_{par}'. $S_p = T_S/T_{par}$
- Efficiency 'E_P' measures the average contribution of each node to the speedup, and is calculated as the ratio $E=S_p/M$. Since $T_{par} \leq T_S$ and $E_P \leq 1$
- Scalability indicates whether a parallel system can improve its performance as its size increase.

To evaluate the performance of the proposed parallel secure video synopsis framework, (9, 18, and 27) videos are processed using 'N' nodes, where each node contains 'M' homogenous machines. Each machine is equipped with an 11th Gen Intel Core i7-11800H Processor running at 2.30 GHz, with 32 GB RAMs. The videos are sorted in descending order of their sequential execution times and placed into a scheduling list, as discussed in Section 2.2. Figures (8, 9, and 10) represent the performance of the proposed model for nine, eighteen, and twenty-seven videos using different number of machines ranging from 2 to 24. Figures (8-a, 9-a, and 10-a) depict the total parallel execution time. Then, figures (8-b, 9-b, and 10-b) show the speedup 'SP'. Moreover, figures (8-c, 9-c, and 10-c) demonstrate the system efficiency 'EP'. Finally, figures (8-d, 9-d, and 10-d) illustrate the improvement of the proposed model compared to the sequential one. Additionally, Figure 11 denotes the effect of increasing the number of videos on the system performance. The experimental results show that the proposed model effectively leverages the parallelism within the query-based video synopsis system [23], to reduce its computational time. Furthermore, this parallel approach can be easily extended and applied to enhance other video synopsis models. The key remarks from the experiments are as follows:

- The maximum number of nodes that achieve a balanced workload for the course-grained level is four. The values of 'M' tested inside each node are 1, 2, 3, 4, 5, and 6, leading to a total number of machines tested of 4, 8, 12, 16, 20, and 24.
- Increasing the number of machines decreases the execution time, as depicted in figures (8-a, 9-a, and 10-a). When the number of machines increases from one to two, the execution time reduces by half. In case of, the number of machines per node equals four (the total number of machines equals 16), the total execution time decreases by around 92% from the sequential time. Using twenty machines (five machines per node) leads to a 93.38% reduction in the sequential processing time. Further, increasing the machines from four to six per node (for the total number rises from 16 to 24) reduces the execution time to 0.057 of the sequential time. As the number of machines increases, the speedup increases as shown in figures (8-b, 9-b, and 10-b). The system efficiency decreases as depicted in figures (8-c, 9-c, and 10-c).
- When the number of machines increases to twenty machines (five machines per node) the efficiency decreases to approximately 75.5%, as shown in figures (8-c, 9-c, and 10-c).
- When increasing the number of machines inside each node to six (total number of machines equals twenty-four), the execution time reduces by 94.2% (very small improvement regarding using twenty machines). This increase in the number of machines decreases the overall system performance (approximately 73.1% efficiency) and requires higher cost. Therefore, a configuration of four nodes each of five machines (total number is twenty) is a suitable compromise between system performance and cost.
- Figure 11 shows that the proposed model is scalable increasing the number of videos and machines can improve the system performance. This is important when executing hundreds or thousands of videos.

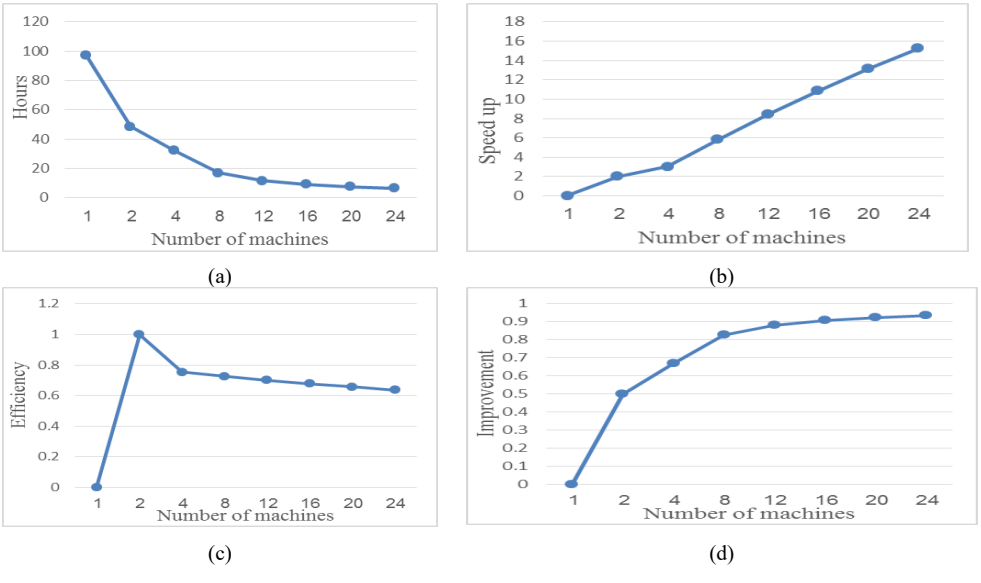


Figure 8: The Performance Analysis of The Distributed Proposed Model When V=9: (a) Execution Time. (b) Speed up (c) Efficiency (d) Improvement Degree

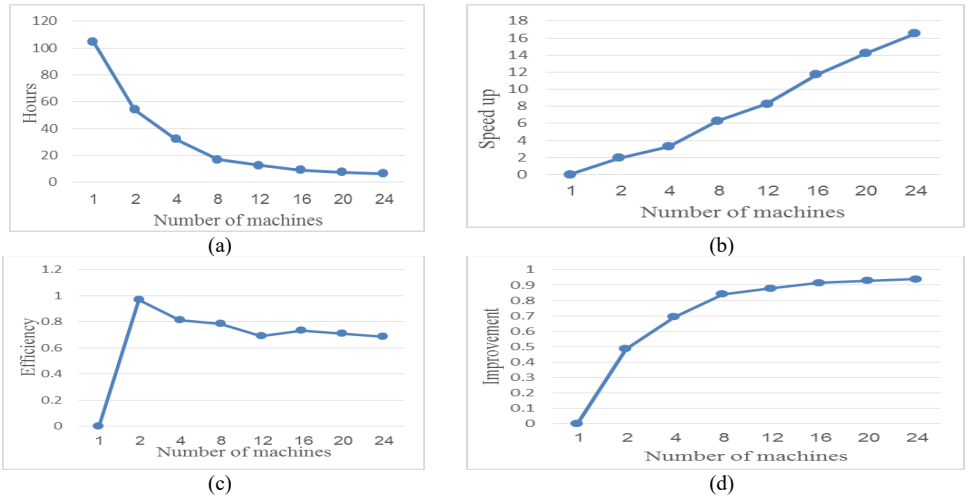
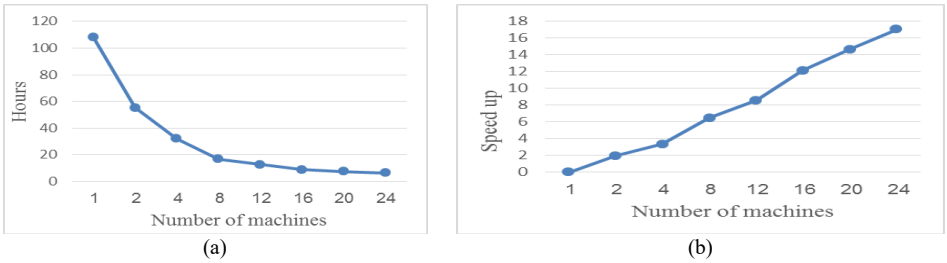
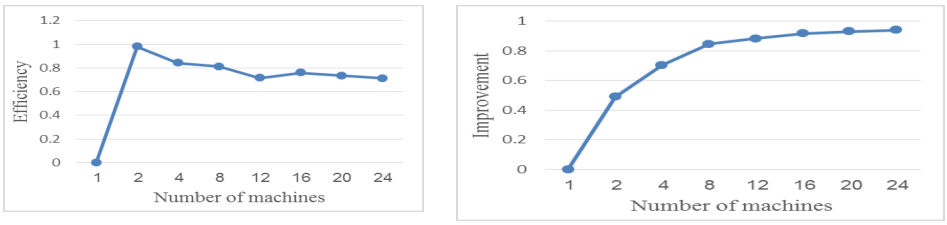


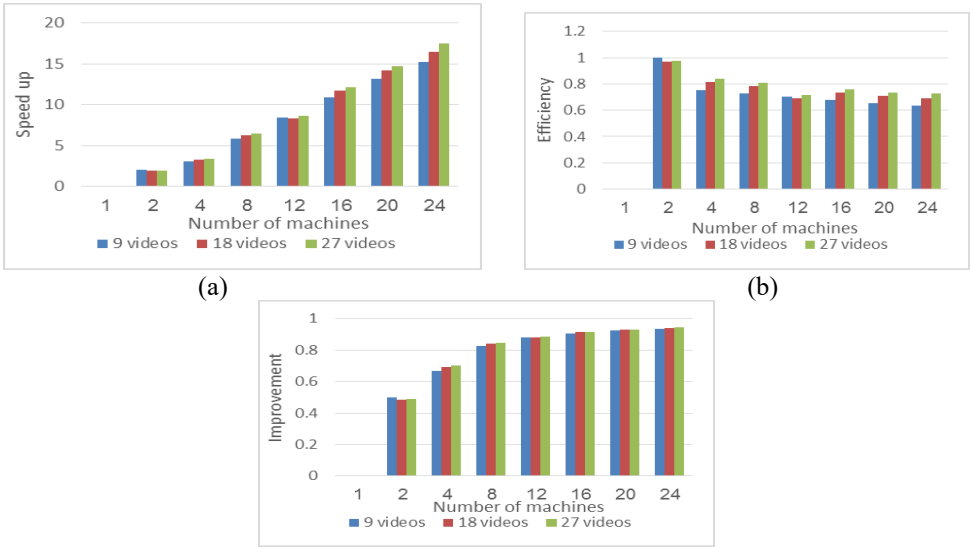
Figure 9. The performance analysis of the distributed proposed model when V=18: (a) Execution Time. (b) Speed up. (c) Efficiency. (d) Improvement Degree





(c) (d)

Figure 10. The Performance Analysis of The Distributed Proposed Model When V=27: (a) Execution Time. (b) Speed up. (c) Efficiency. (d) Improvement Degree



(a) (b)

(c)

Figure 11: Impact of Video Quantity on Performance Metrics: (a) Speedup, (b) Efficiency, (c) Improvement Degree

4. CONCLUSIONS

This research paper is concerned with two directions. The first is proposing a video synopsis mechanism to detect intrusion behavior using a query-based approach. The intrusion detection focuses on recognizing unauthorized actions and the visual appearance of individuals, allowing users to define and describe these illicit activities. The proposed algorithm is developed based on a conventional query-based method and improves its computational efficiency by employing a distributed system, resulting in a significant reduction in processing time; and that is the second direction addressed by the paper. The Parallelism is handled at the inter-video level (coarse-grained parallelism) and the intra-video level (fine-grained parallelism) to maximize overall performance. Despite the advantages offered by distributed systems, they pose security concerns, particularly the risk of person tube substitution and tube loss during transfers between devices. Therefore, the proposed system incorporates a two-layer security model to safeguard the object's tubes against loss or unauthorized insertion, thereby preserving the integrity of the entire process. The first layer utilizes a watermark technique to immune the system against tube substitution. Meanwhile, the second layer employs a serialization technique to protect the tubes from being lost. The results prove that the proposed system is a high-quality and secured video synopsis system that performs efficiently in computational time.

As a limitation of the proposed system, although it resists some intrusion attacks well, but a more qualified security system may be needed to resist sophisticated attacks. On the other hand, the system addresses predefined queries and does not adapt unstructured or novel queries. These two points may be addressed as future work.

ACKNOWLEDGMENT

This paper is supported by Korean Agency for Technology and Standard under Ministry of Trade, Industry and Energy in 2023, project numbers are 1415181629 (Development of International Standard Technologies based on AI Learning and Inference Technologies), 1415181629 (Development of International Standard Technologies based on AI Model Lightweighting Technologies), and 1415181638 (Establishment of standardization basis for BCI and AI Interoperability).

REFERENCES

[1] A. Rav-Acha, Y. Pritch, and S. Peleg, "Making a long video short: Dynamic video synopsis," *Proceedings of IEEE Computer Socitey Conference Computer Vision and Pattern Recognition*, 17-22 June, New York, NY, USA, Vol. 1, 2006, pp. 435–441.

[2] K. Baskurt and R. Samet, "Video synopsis: A survey," *Computer Vision and Image Understanding*, Vol. 181, No. December 2017, pp. 26–38, doi: 10.1016/j.cviu.2019.02.004.

[3] S. Gandhi and T. Ratanpara, "Surveillance Video Synopsis Techniques : A Review," International Journal of Next-Generation Computing (IJNGC), Vol. 8, Issue. 3, 2017.

[4] M. Moussa and R. Shoitan, "Object-based video synopsis approach using particle swarm optimization," *Signal, Image and Video Processing*, Vol. 15, No. 4, 2021, pp. 761–768. doi: 10.1007/s11760-020-01794-1.

[5] Y. Nie, H. Sun, P. Li, C. Xiao, and K. Ma, "Object Movements Synopsis via Part Assembling and Stitching," *IEEE Transactions on Visualization Computer Graphics*, Vol. 20, No. 9, 2013, pp. 1303–1315.

[6] Y. Nie et al., "Collision-Free Video Synopsis Incorporating Object Speed and Size Changes," *IEEE Transactions on Image Processing*, Vol. 29, 2019, pp. 1465-1478.

[7] B. Raman, S. Kumar, P. Roy, and D. Sen, "Surveillance Video Synopsis While Preserving Object Motion Structure and Interaction," *Proceedings of International Conference on Computer Vision and Image Processing. Advances in Intelligent Systems and Computing*, Vol. 460, 2016, pp. 197-207.

[8] S. Ghatak, S. Rup, B. Majhi, and M. Swamy, "An improved surveillance video synopsis framework: a HSATLBO optimization approach," *Multimedim Tools Applications*, Vol. 97, 2019, pp. 4429-4461. doi: 10.1007/s11042-019-7389-7.

[9] S. Ghatak and S. Rup, "Performance Study of Some Recent Optimization Techniques for Energy Minimization in Surveillance Video Synopsis Framework", *Information, Photonics and Communications*, LNSS Vol. 79, Springer Singapore, 2020, pp. 227-237.

[10] S. Ghatak, S. Rup, B. Majhi, and M. Swamy, "HSAJAYA: An Improved Optimization Scheme for Consumer Surveillance Video Synopsis Generation," *IEEE Transactions Consum. Electronics*, Vol. 66, No. 2, 2020, pp. 144–152. doi: 10.1109/TCE.2020.2981829.

[11] T. Yao, M. Xiao, C. Ma, C. Shen, and P. Li, "Object based video synopsis," *Proceedings of IEEE Workshope Advanced Research and Technoogy in Industry Applications,* Otawa, ON, 29-30 Sept., 2014, pp. 1138–1141, 2014.

[12] X. Li, Z. Wang, and X. Lu, "Surveillance video synopsis via scaling down objects," *IEEE Transanactions on Image Processing*, Vol. 25, No. 2, pp. 740–755, 2016, doi: 10.1109/TIP.2015.2507942.

[13] C. Wang et al., "Video synopsis method based on interaction determination using human pose estimation," *Journal of Electronics Imaging*, Vol. 33, No. 01, 2024, doi: 10.1117/1.jei.33.1.013053.

[14] K. Namitha, M. Geetha, and N. Athi, "An Improved Interaction Estimation and Optimization Method for Surveillance Video Synopsis," *IEEE Multimedia*, Vol. 30, No. 3, 2023, pp. 25–36. doi: 10.1109/MMUL.2022.3224874.

[15] S. Feng, Z. Lei, D. Yi, and S. Li, "Online content-aware video condensation," *Proceedings of IEEE Computer Society Conference of Computer Vision Pattern Recognition*, Providence, USA, No. 1, 16-21 June, 2012, pp. 2082–2087.

[16] K. Başkurt and R. Samet, "Long-term multiobject tracking using alternative correlation filters," *Turkish Journal Electrical Engineeing and Computer Scince*, Vol. 26, No. 5, 2018, pp. 2246–2259. doi: 10.3906/elk-1709-245.

[17] M. Lu, Y. Wang, and G. Pan, "Generating fluent tubes in video synopsis," *Proceedings of* the *IEEE International Confernce Acoustics, Speech, and Signal Processin*, Vancover, BC, Canada, 26-31 May 2013, pp. 2292–2296.

[18] L. Lin, W. Lin, W. Xiao, and S. Huang, "An optimized video synopsis algorithm and its distributed processing model," *Software Computing*, Vol. 21, No. 4, 2017, pp. 935–947. doi: 10.1007/s00500-015-1823-1.

[19] S. Ahmed et al., "Query-Based Video Synopsis for Intelligent Traffic Monitoring Applications," *IEEE Transactions on Intelligence Transrtation. Systems*, Vol. 12, issue 8, 2019, pp. 1–12. doi: 10.1109/ tits.2019.2929618.

[20] K. Namitha, A. Narayanan, and M. Geetha, "Interactive visualization-based surveillance video synopsis," *Applied . Intelligence*, Vol. 52, 2021, pp. 3954-3975. doi: 10.1007/s10489-021-02636-4.

[21] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 11, 2008, pp. 1971–1984. doi: 10.1109/TPAMI.2008.29.

[22] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg, "Clustered synopsis of surveillance video," *Proceedings of the 6th IEEE Inernational Conference on Advanced Video Signal Based Surveillance, AVSS 2009*, Genova, Italy, 02-04 Sept. 2009, pp. 195–200.

[23] M. Abdalah et al., "User Preference-Based Video Synopsis Using Person Appearance and Motion Descriptions," *Sensors,* Vol. 23, No. 3, 1521, (Switzerland), 2023.

[24] D. Cochard, "ByteTrack : Tracking model that also considers low accuracy bounding boxes," [Online]: arXiv:2110.06864, October 2021.

[25] Y. Zhang et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," [Online]. Available: *arXiv*, 2021.

[26] N. Agarwal, A. Singh, and P. Singh, "Survey of robust and imperceptible watermarking," *Multimeda Tools and Applications*, Vol. 78, No. 7, Apr. 2019, pp. 8603–8633. doi: 10.1007/s11042-018-7128-5.

[27] B. Chen and G. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding,", *IEEE Information Theory*, Vol.47, No. 4, 2001, pp. 1423-1443.

[28] B. Borah, Sagarika, "Quantization Index Modulation (QIM) based watermarking techniques for 3D meshes," *Proceedings of the fourth International Conference of Image Information Processing (ICIIP)*, Shimla, India, 21-23 December 2017.

[29] J. Mao, H. Tang, S. Lyu, Z. Zhou, and X. Cao, "Content-Aware Quantization Index Modulation: Leveraging Data Statistics for Enhanced Image Watermarking," Jun. 2023, [Online]. Available: arXiv.org/abs/2306.15896.

[30] H. El-Rewini and M. Abd-El-Barr, Advanced Computer Architecture and Parallel Processing. Book: © John Weilly & Sons Inc., 2005.

[31] J. Hennessy, Computer architecture: a quantitative approach. Book: © Elsevier, 2012.

[32] R. Trobec, B. Slivnik, P. Bulić, and B. Robič, Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-art Platforms. Book © Springer, 2018.

[33] G. Elkabbany and M. Moussa, "Accelerating video encoding using cluster computing," *Multimedia Tools and Appications*, Vol. 79, No. March 2019, 2020, pp. 17427-17444. doi: 10.1007/s11042-020-08717-9.

[34] I. Ahmad, Y. He, and M. Liou, "Video compression with parallel processing," *Parallel Computing*, Vol. 28, No. 7–8, 2002, pp. 1039–1078. doi: 10.1016/S0167-8191(02)00100-X.

[35] Z. Chang, B. Jong, W. Wong, and M. Wong, "Distributed video transcoding on a heterogeneous computing platform," *Proceedings of the 2016 IEEE Asia Pacific Conference on Circuits Systems (APCCAS)*, Jeju, Korea (South), 25-28 Oct. 2016, pp. 444–447.

[36] C. Zhong, X. Jiang, and G. Qi, "Video-based Person Re-identification Based on Distributed Cloud Computing," *Journal of Artificial Intelligence Technology*, Vol. 1, No. 2, 2021, pp. 110–120. doi: 10.37965/jait.2020.0058.

[37] B. Wilkinson, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2/E. Book: © Pearson Education, 2006.

[38] G. El Kabbany, N. Wanas, N. Hegazi, and S. Shaheen, "A dynamic load balancing framework for real-time applications in message passing systems," *International Journal of Parallel Programing*, Vol. 39, No. 2, 2011, pp. 143–182. doi: 10.1007/s10766-010-0134-5.

[39] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," *Proceedings of IEEE Computer Socitey Confernce Computer Vision and Pattern Recognition*, Colorado, CO, USA, 20-25 June 2011, pp. 3457–3464.

[40] "The VIRAT Video Dataset." https://viratdata.org/.

[41] A. Sivasubramaniam, A. Singla, U. Ramachandran, and H. Venkateswaran, "An application-driven study of parallel system overheads and network bandwidth requirements," *IEEE Tranactions of Parallel Distriuted Systems*, Vol. 10, No. 3, 1999, pp. 193–210. doi: 10.1109/71.755819.

[42] M. Usman *et al.*, "Error Concealment for Cloud-Based and Scalable Video Coding of HD Videos," *IEEE Tranations of Cloud Computing*, Vol. 7, No. 4, 2019, pp. 975–987, doi: 10.1109/TCC.2017.2734650.