SMART MEDIA VIDEO CLOUD TECHNOLOGY IN NEWS COMMUNICATION

Reference NO. IJME 1350, DOI: 10.5750/ijme.v1i1.1350

YY Zhang*, Department of Humanities and Media, Yangtze University College of Arts and Sciences, Jingzhou, Hubei, 434020, China

* Corresponding author. YY Zhang (Email): xiaosheng5h21@163.com

KEY DATES: Submission date: 20.12.2023 / Final acceptance date: 27.02.2024 / Published date: 12.07.2024

SUMMARY

Cloud technology revolutionizes the way businesses and individuals store, access, and manage data and applications. It involves the use of remote servers hosted on the internet to store and process information, providing on-demand computing resources. With the cloud, users can access their data and applications from anywhere with an internet connection, fostering increased collaboration, flexibility, and scalability. This research explores the convergence of smart media video cloud technology and news communication, introducing a novel framework named Big Data Analytics Parallel Edge Computing with ECC (PEC-ECC). With the proliferation of video content in news dissemination, there is a growing need for innovative technologies to enhance efficiency, security, and accessibility. PEC-ECC integrates the power of big data analytics, parallel edge computing, and error-correcting code mechanisms to optimize video processing and delivery. This framework not only addresses the challenges of data volume and computational speed but also ensures the integrity and security of transmitted content. By leveraging edge computing capabilities, PEC-ECC minimizes latency, making real-time news communication more responsive and reliable. This research contributes to the advancement of smart media technology, offering a robust solution to elevate the quality and effectiveness of video-based news communication in the era of digital journalism.

KEYWORDS

Cloud Computing, News Communication, Video communication, Edge Computing, Big Data Analytics

NOMENCLATURE

PEC-ECC	Parallel Edge Computing with Elliptic Curve Cryptography
ECC	Elliptic Curve Cryptography
F	Frequency

1. INTRODUCTION

Automatic numbering systems must not be used. Cloud computing has revolutionized the landscape of modern technology by providing a flexible and scalable framework for storing, managing, and accessing data and applications over the internet [1]. Unlike traditional on-premises infrastructure, cloud computing offers organizations the ability to utilize computing resources on-demand, paying only for what they use, without the need for extensive hardware investments or maintenance. This paradigm shift has empowered businesses to streamline operations, enhance collaboration, and innovate more rapidly, as resources can be provisioned and scaled dynamically to meet changing demands [2]. With its promise of cost-efficiency, resilience, and accessibility, cloud computing has become an integral component of the digital ecosystem, driving efficiency, agility, and competitiveness across industries [3]. Cloud computing has synergized seamlessly with big data analytics, marking a pivotal advancement in the realm of data-driven decision-making. By leveraging the vast computational power and storage capabilities of the cloud, organizations can efficiently process, analyze, and derive actionable insights from massive volumes of structured and unstructured data [4]. This amalgamation not only enables businesses to extract valuable information but also enhances scalability, allowing for the processing of data at unprecedented scales without the constraints of on-premises infrastructure. Moreover, cloud-based big data analytics solutions offer agility and flexibility, facilitating rapid experimentation and iteration in developing analytical models [5]. As a result, businesses can uncover hidden patterns, trends, and correlations in their data, empowering them to make informed decisions, optimize operations, and gain a competitive edge in today's data-driven landscape [6]. Big data analytics, coupled with smart media video cloud technology, is revolutionizing news communication in profound ways. This innovative fusion enables news organizations to harness the power of big data to analyze audience preferences, consumption patterns, and engagement metrics, thereby tailoring

content delivery to specific demographics and interests [7]. With leveraging cloud-based video infrastructure, news outlets can efficiently store, manage, and distribute multimedia content across various platforms, ensuring seamless access for viewers worldwide. Furthermore, advanced analytics algorithms sift through vast amounts of data generated by viewer interactions, allowing news broadcasters to personalize content recommendations, optimize ad placements, and enhance user experiences [8]. This dynamic integration of big data analytics with smart media video cloud technology not only enhances the relevance and timeliness of news reporting but also fosters deeper audience engagement and loyalty in today's rapidly evolving media landscape [9].

The integration of big data analytics with smart media video cloud technology brings a multitude of benefits to the realm of news communication [10]. Firstly, big data analytics allows news organizations to gain deep insights into audience behavior and preferences. By analyzing data points such as viewership patterns, content consumption habits, and engagement metrics, news broadcasters can understand what content resonates most with their audience [11]. This insight is invaluable for tailoring content production and distribution strategies to meet the specific interests and needs of viewers, ultimately leading to higher engagement and retention rates [12]. Furthermore, the use of smart media video cloud technology facilitates seamless content delivery across various platforms and devices [13]. Cloud-based infrastructure enables news outlets to store and manage vast libraries of multimedia content efficiently, ensuring accessibility and scalability [14]. This means that viewers can access news updates, live streams, and archived content from anywhere, at any time, enhancing the reach and impact of journalistic endeavors [15]. Moreover, the combination of big data analytics and cloud technology enables dynamic content personalization. By leveraging data-driven algorithms, news broadcasters can deliver personalized recommendations and curated content experiences to individual viewers [16]. This level of customization not only enhances user satisfaction but also increases the likelihood of continued engagement and loyalty. Additionally, the integration of big data analytics into news communication facilitates data-driven decision-making across various aspects of content production and distribution [17]. From editorial planning and story selection to advertisement placement and audience targeting, data-driven insights empower news organizations to optimize their strategies for maximum impact and relevance.

The paper makes several significant contributions to the field of edge computing and cryptographic techniques, particularly in the context of smart media for news communication. The paper introduces a novel framework, Parallel Edge Computing with Elliptic Curve Cryptography (PEC-ECC), which integrates parallel computing techniques with ECC for secure and efficient processing of smart media data at the network edge. This framework addresses the need for robust solutions capable of handling the increasing volumes of multimedia content in real-time communication systems. With ECC for data encryption and decryption, the PEC-ECC framework offers enhanced security features, ensuring data confidentiality and integrity during transmission and processing. This contributes to strengthening the security posture of edge computing environments, particularly in scenarios where sensitive information, such as news content, is involved. PEC-ECC enables real-time analytics of smart media data, allowing for timely extraction of insights and actionable information from large datasets. This capability is essential for news communication applications, where rapid analysis of multimedia content is crucial for delivering up-to-date and relevant information to users. Through experimentation and analysis, the paper evaluates the cost implications of employing PEC-ECC for smart media applications. By optimizing resource utilization and minimizing processing overhead, PEC-ECC offers a cost-effective solution for processing and transmitting multimedia content in edge computing environments. The scalability and versatility of the PEC-ECC framework are demonstrated across various scenarios, including video streams, audio recordings, images, and text articles. This highlights its adaptability to different types of smart media data and its potential for broader application in diverse domains beyond news communication. The contributions of the paper lie in its development of an innovative framework that addresses key challenges in edge computing, particularly in the context of handling multimedia content securely and efficiently. By providing a comprehensive analysis and evaluation of the PEC-ECC framework, the paper contributes valuable insights and solutions to advance the capabilities of edge computing systems in processing smart media for news communication and beyond.

2. CLOUD COMPUTING IN SMART MEDIA

Cloud computing plays a pivotal role in enabling smart media applications to operate efficiently and effectively. Smart media encompasses various forms of digital content, including audio, video, images, and interactive media, which are delivered and consumed across a wide range of devices and platforms. Cloud computing provides the underlying infrastructure and services necessary to store, process, and deliver smart media content to users worldwide. One key aspect of cloud computing in smart media is the provision of scalable and reliable storage solutions. Cloud storage services, such as Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage, offer virtually unlimited capacity for storing large volumes of media files. The storage capacity can be easily scaled up or down based on demand, ensuring that smart media applications can accommodate fluctuating storage requirements efficiently. In addition to storage, cloud computing also provides powerful computational resources for processing smart media content. This includes tasks such as transcoding,

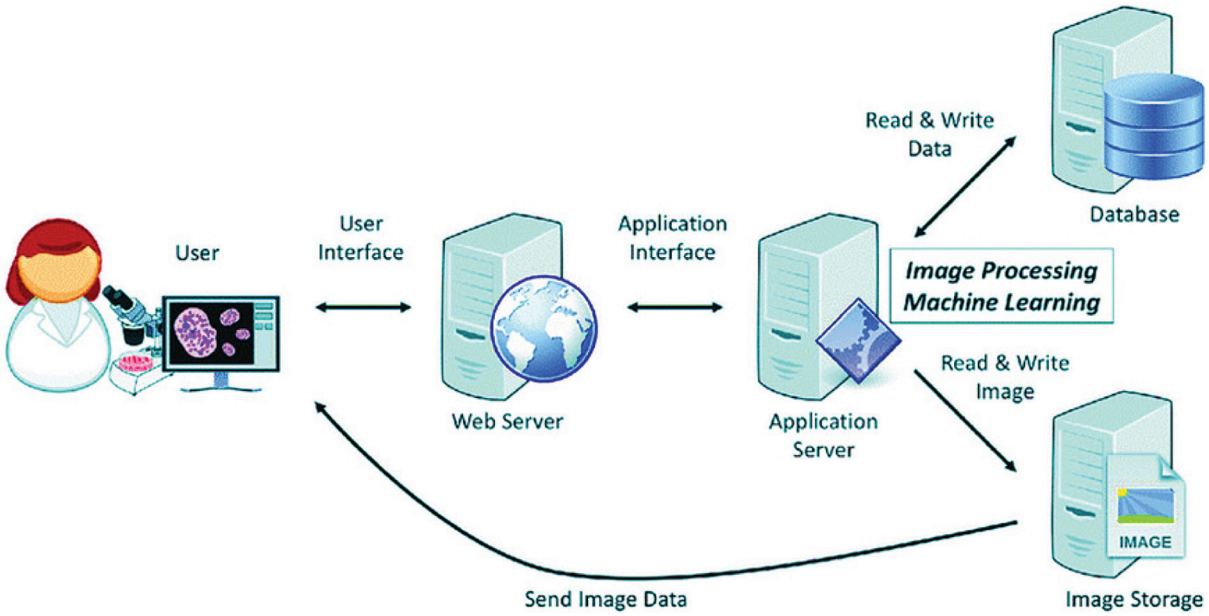

Figure 1. Cloud technology with PEC-ECC

encoding, decoding, and rendering, which are essential for delivering high-quality media experiences to users. The figure 1 illustrated the cloud technology for the PEC-ECC model environment.

Cloud-based media processing services, such as Amazon Elastic Transcoder or Google Cloud Media Translation, leverage distributed computing resources to process media files quickly and cost-effectively. Furthermore, cloud computing enables smart media applications to leverage advanced analytics and machine learning algorithms for content recommendation, personalization, and optimization. By analyzing user behavior, preferences, and engagement metrics, smart media platforms can deliver targeted content recommendations and personalized experiences to individual users. Cloud-based analytics services, such as Amazon Personalize or Google Cloud AI Platform, provide the tools and infrastructure necessary to build and deploy machine learning models at scale. The cost of using cloud computing services for smart media applications can be represented in equation (1)

$$C = S + P + T \tag{1}$$

In equation (1) C represents the total cost of cloud computing services; S represents the cost of storage, including the price per unit of storage and any additional fees for data transfer or retrieval; P represents the cost of processing, including the price per unit of computational resources (e.g., CPU hours, GPU hours) and any additional fees for specialized services (e.g., media transcoding, machine learning inference). T represents the cost of data transfer, including the price per unit of data transfer between the cloud provider and external networks or regions. Cloud computing revolutionizes smart media by providing scalable, on-demand access to resources such as storage, processing power, and analytics tools. This scalability is particularly crucial for smart media applications, which often experience fluctuating demands in terms of storage capacity, processing requirements, and user traffic. Storage: Cloud storage services offer virtually limitless capacity for storing smart media content. These services typically charge based on the amount of data stored and any additional features such as data redundancy, access control, or data lifecycle management. The cost of storage S can be calculated with equation (2)

$$S = N \times P_s + T_s \tag{2}$$

In equation (2) N is the total amount of data stored; Ps is the price per unit of storage; and Ts represents any additional charges for data transfer, retrieval, or other storage-related operations. Cloud computing platforms provide powerful computational resources for processing smart media content. This includes tasks such as transcoding (converting media files from one format to another), encoding (compressing media files for efficient storage and transmission), decoding (reversing the compression process for playback), and rendering (generating visual effects or overlays). The cost of processing P can be calculated using the equation (3)

$$P = N \times C_p \tag{3}$$

In equation (3) N is the total amount of data processed and C_p is the price per unit of computational resources (e.g., CPU hours, GPU hours). Cloud providers typically charge for data transferred into and out of their networks, as well as between different regions or availability zones. This can include data transfer between storage and processing resources, as well as between the cloud provider's network and external networks (e.g., the internet). The cost of data transfer (T) can be calculated using the equation (4)

$$T = N \times P_t \tag{4}$$

In equation (4) N is the total amount of data transferred and Pt is the price per unit of data transfer. To minimize the overall cost of cloud computing services (C), smart media applications must optimize their usage of storage, processing, and data transfer resources. This involves careful consideration of factors such as data storage strategies (e.g., tiered storage, data compression), computational efficiency (e.g., task parallelization, resource allocation), and data transfer optimization (e.g., caching, content delivery networks).

3. SYSTEM MODEL

The system model for Big Data Analytics Parallel Edge Computing with ECC (PEC-ECC) encompasses several key components and concepts. Big Data Analytics is the process of analyzing large and complex datasets to uncover hidden patterns, correlations, and insights. Big data analytics techniques may include data mining, machine learning, statistical analysis, and visualization. Parallel Edge Computing (PEC) involves distributing computing tasks across a network of edge devices located close to the data source or end-users. This decentralized approach reduces latency and bandwidth usage by processing data locally, improving overall system efficiency and responsiveness. Elliptic Curve Cryptography (ECC) algorithm based on elliptic curves over finite fields. It offers strong security with shorter key lengths compared to traditional cryptographic algorithms such as RSA. ECC is well-suited for resource-constrained environments like edge devices due to its efficiency. PEC-ECC system architecture comprises edge devices, communication networks, and centralized or distributed servers for data processing and storage. Edge devices, such as IoT sensors, smartphones, or edge servers, perform local data processing and ECC-based encryption before transmitting data to the central servers. Data generated at the edge devices are processed locally using parallel computing techniques. ECC is applied to encrypt sensitive data before transmission over the network to ensure confidentiality and integrity. Encrypted data is then sent to centralized or distributed servers for further analysis and storage. The system leverages parallel computing paradigms, such as parallel algorithms and distributed computing frameworks (e.g., MapReduce, Apache Spark), to efficiently process large volumes of data in parallel across multiple edge devices. This parallelism enables faster data processing and real-time analytics, enhancing system performance and scalability. ECC is employed to provide secure communication and data encryption between edge devices and central servers. By encrypting data at the edge before transmission, the system ensures data confidentiality and integrity, protecting against unauthorized access and cyber-attacks. The distributed nature of PEC-ECC allows the system to scale dynamically by adding or removing edge devices as needed. This scalability improves system efficiency by distributing computational load and reducing network congestion, enabling the system to handle increasing volumes of data and users.

The system model for Big Data Analytics Parallel Edge Computing with ECC (PEC-ECC) is designed to optimize data processing, security, and efficiency in edge computing environments. This model combines the parallel processing capabilities of edge devices with the cryptographic strength of Elliptic Curve Cryptography (ECC) to enable secure and scalable big data analytics at the network edge. The PEC-ECC system model is the utilization of parallel computing techniques to distribute computational tasks across multiple edge devices. This parallelism enhances the system's ability to process large volumes of data in real-time, leveraging the computational resources available at the edge. Mathematically, the parallelism can be represented using parallel algorithms and distributed computing frameworks, such as MapReduce, which enable efficient data processing and analysis across distributed nodes. Additionally, ECC is employed to provide robust security for data transmission and storage within the system. The cryptographic strength of ECC allows for the encryption of sensitive data at the edge devices before transmission over the network. The encryption process involves the derivation of elliptic curve parameters and the generation of public-private key pairs, which are used to encrypt and decrypt data. Mathematically, the ECC encryption process can be represented by the elliptic curve equation and the point addition and scalar multiplication operations defined over finite fields. The integration of parallel edge computing and ECC within the PEC-ECC system model offers several advantages. Firstly, it enhances data security by ensuring the confidentiality and integrity of data transmitted between edge devices and central servers. Secondly, it improves system efficiency by enabling parallel processing of data at the network edge, reducing latency and bandwidth usage. Lastly, it enhances scalability by allowing the system to dynamically adapt to changing workload demands and resource availability.

4. PROPOSED BIG DATA ANALYTICS PARALLEL EDGE COMPUTING WITH ECC (PEC-ECC)

The proposed Big Data Analytics Parallel Edge Computing with ECC (PEC-ECC) system aims to revolutionize data processing, security, and efficiency in edge computing environments. This innovative framework integrates parallel edge computing techniques with Elliptic Curve Cryptography (ECC) to enable secure and scalable big data analytics at the network edge. The system architecture of PEC-ECC entails the deployment of edge devices equipped with computational resources and ECC-enabled encryption capabilities. These edge devices operate in parallel, leveraging distributed computing techniques to process data locally and in real-time. Mathematically, the parallelism in PEC-ECC can be represented using parallel algorithms and distributed computing frameworks such

as MapReduce, which facilitate efficient data processing and analysis across distributed nodes. Furthermore, ECC is employed to ensure robust security for data transmission and storage within the system. The cryptographic strength of ECC enables the encryption of sensitive data at the edge devices before transmission over the network. This encryption process involves the derivation of elliptic curve parameters and the generation of public-private key pairs, which are used for encryption and decryption operations. Mathematically, the ECC encryption process can be represented by the elliptic curve equation and the point addition and scalar multiplication operations defined over finite fields.

The integration of parallel edge computing and ECC within the PEC-ECC system offers several key advantages. Firstly, it enhances data security by providing confidentiality and integrity for data transmitted between edge devices and central servers. Secondly, it improves system efficiency by enabling parallel processing of data at the network edge, reducing latency and bandwidth usage. Lastly, it enhances scalability by allowing the system to dynamically adapt to changing workload demands and resource availability. In parallel edge computing, tasks are distributed across multiple edge devices for concurrent processing, improving overall system performance. Mathematically, parallelism can be represented using parallel algorithms, such as parallel matrix multiplication or parallel sorting algorithms. For example, in parallel matrix multiplication, if A and B are two matrices to be multiplied, and C is the resulting matrix, the parallel algorithm could distribute rows or columns of A and B across multiple edge devices, each computing a portion of the result C using equation (5)

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \times B_{kj} \tag{5}$$

In equation (5) *Cij* represents the ith row and jth column of the resulting matrix *C*, and *n* is the size of the matrices. ECC is a cryptographic algorithm based on elliptic curves over finite fields. It offers strong security with shorter key lengths compared to traditional cryptographic algorithms such as RSA. The core operation in ECC is point addition and scalar multiplication over elliptic curves. The elliptic curve equation is given in equation (6)

$$y^2 \equiv x^3 + ax + b \left(mod\ p \right) \tag{6}$$

In equation (6) *a* and *b* are constants defining the curve, and *p* is a prime number representing the field size. Point addition and scalar multiplication operations in ECC involve manipulating points on the elliptic curve according to specific rules. For example, point addition involves finding the third point on the curve that lies on the line passing through two given points, while scalar multiplication involves repeatedly adding a point to itself a certain number of times denoted in equation (7) and equation (8)

$$P_3 = P_1 + P_2 \tag{7}$$

$$Q = kP \tag{8}$$

In equation (7) P_3, P_1, P_2, are points on the curve, k is a scalar (private key), and Q is the resulting point (public key).

Algorithm 1: PEC-ECC for the Smar Media

```
function PEC_ECC_Processing(data):
    // Parallel Edge Computing (PEC) at the edge devices
    parallel_process(data) // Distribute data processing tasks
across edge devices
    // Elliptic Curve Cryptography (ECC) for data encryption
    key_pair = generate_key_pair() // Generate ECC public-
private key pair
    for each data_chunk in processed_data:
        encrypted_chunk = ECC_encrypt(data_chunk, public_
key) // Encrypt data using ECC
    send_encrypted_data(encrypted_data) // Transmit
encrypted data to central server
function parallel_process(data):
    // Parallel processing of data across multiple edge devices
    divide_data(data) // Divide data into smaller chunks
    for each chunk in data_chunks:
        process_chunk(chunk) // Process data chunk in parallel
function ECC_encrypt(data_chunk, public_key):
    // ECC encryption of data using public key
    encrypted_data_chunk = ECC_encrypt(data_chunk,
public_key)
    return encrypted_data_chunk
function generate_key_pair():
    // Generate ECC public-private key pair
    private_key = ECC_generate_private_key()
    public_key = ECC_generate_public_key(private_key)
    return (private_key, public_key)
function send_encrypted_data(encrypted_data):
    // Transmit encrypted data to central server
    establish_connection() // Establish secure connection with
central server
    send_data(encrypted_data) // Send encrypted data over the
network
    close_connection() // Close connection with central server
```

4.1 CRYPTOGRAPHIC PROCESS WITH PEC-ECC FOR THE SMART MEDIA FOR NEWS COMMUNCIATION

In the context of smart media for news communication, the integration of Cryptographic Process with PEC-ECC (Parallel Edge Computing with Elliptic Curve Cryptography) offers a robust framework for ensuring the security and privacy of sensitive data, while enabling efficient data processing at the network edge. The cryptographic process begins with the generation of ECC key pairs, comprising a public key for encryption and a corresponding private key for decryption. Mathematically, the ECC key pair generation involves selecting a random private key (k) and computing the corresponding public key ($Q = kG$), where G represents a generator point on

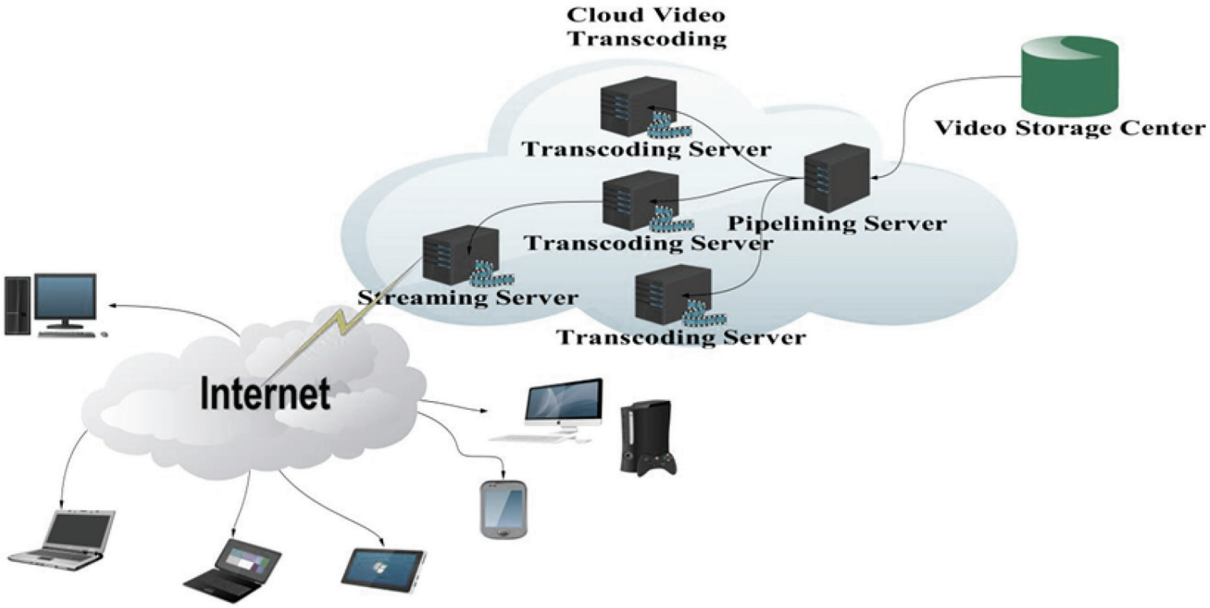

Figure 2. Cloud server for the PEC-ECC

the elliptic curve. This process can be represented by the equation using equation (9)

$$Q = k \times G \tag{9}$$

Once the key pair is generated, the public key is distributed to edge devices, while the private key is securely maintained by the central server. In the PEC-ECC framework, edge devices perform parallel processing of smart media data, such as video streams or textual content, using parallel computing techniques. This parallelism enhances the efficiency of data processing, enabling real-time analysis and content delivery. Mathematically, parallel processing involves distributing data chunks across multiple edge devices and executing processing tasks concurrently, optimizing computational resources and reducing processing latency. Simultaneously, the cryptographic process ensures the confidentiality and integrity of smart media data during transmission from edge devices to the central server. Each data chunk processed at the edge is encrypted using the ECC public key before being transmitted over the network. The ECC encryption process involves mapping the plaintext data onto points on the elliptic curve and computing ciphertext points using scalar multiplication. Mathematically, ECC encryption can be represented in equation (10)

$$C = M + k \times Q \tag{10}$$

In equation (10) M represents the plaintext data, C represents the ciphertext, Q is the ECC public key, and k is a randomly generated scalar. Upon receiving the encrypted data, the central server utilizes its private key to decrypt the ciphertext and perform further analysis or distribution. This decryption process involves scalar multiplication of the ciphertext with the server's private key, resulting in the retrieval of the original plaintext data. Mathematically, ECC decryption can be represented in equation (11)

$$M = C - d \times Q \tag{11}$$

In equation (11) M represents the decrypted plaintext, C represents the received ciphertext, Q is the ECC public key, and d is the server's private key. Figure 2 illustrated the cloud server environment for the PEC-ECC model in the smart cloud environment.

Algorithm 2: Smar Media with PEC-ECC

```
function PEC_ECC_Processing(media_data):
    // Parallel Edge Computing (PEC) at the edge devices
    parallel_process(media_data) // Distribute data processing tasks across edge devices
    // Elliptic Curve Cryptography (ECC) for data encryption
    key_pair = generate_key_pair() // Generate ECC public-private key pair
    for each data_chunk in processed_media_data:
        encrypted_chunk = ECC_encrypt(data_chunk, public_key) // Encrypt data using ECC
    send_encrypted_data(encrypted_data) // Transmit encrypted data to central server
function parallel_process(media_data):
    // Parallel processing of media data across multiple edge devices
    divide_media_data(media_data) // Divide media data into smaller chunks
        for each chunk in media_data_chunks:
        process_chunk(chunk) // Process media data chunk in parallel
function ECC_encrypt(data_chunk, public_key):
    // ECC encryption of data using public key
    encrypted_data_chunk = ECC_encrypt(data_chunk, public_key)
    return encrypted_data_chunk
function generate_key_pair():
   // Generate ECC public-private key pair
   private_key = ECC_generate_private_key()
```

```
    public_key = ECC_generate_public_key(private_key)
    return (private_key, public_key)
function send_encrypted_data(encrypted_data):
    // Transmit encrypted data to central server
    establish_connection() // Establish secure connection with central server
    send_data(encrypted_data) // Send encrypted data over the network
    close_connection() // Close connection with central server
```

5. EXPERIMENTAL SETUP

The experimental setup for PEC-ECC involves configuring a test environment to evaluate the performance and efficacy of the proposed system in processing smart media data with parallel edge computing and Elliptic Curve Cryptography (ECC). Firstly, a set of edge devices representative of real-world scenarios, such as IoT devices, edge servers, or mobile devices, is selected. These devices should have sufficient computational capabilities to perform data processing tasks in parallel and support ECC-based encryption and decryption. Next, a central server is provisioned to coordinate and manage the processing tasks performed by the edge devices. The server should have adequate resources to handle incoming encrypted data, decrypt it using ECC, and perform further analysis or distribution as required. For the experimental data, a diverse set of smart media content is chosen, including video streams, audio recordings, images, and text articles. This data should represent typical content encountered in news communication scenarios, varying in size, format, and complexity. The ECC parameters, such as the elliptic curve equation, prime modulus, and generator point, are predefined and shared among the edge devices and central server for encryption and decryption operations. The experimental setup also includes establishing communication channels between the edge devices and the central server. Secure communication protocols, such as HTTPS or TLS, are employed to ensure the confidentiality and integrity of data transmission. To evaluate the performance of PEC-ECC, various metrics are considered, including processing latency, throughput, resource utilization, and energy efficiency. These metrics are measured under different scenarios, such as varying data sizes, computational loads, and network conditions, to assess the scalability and robustness of the system. Table 1 shows simalation setup.

Table 1. Simulation setup

Component	Description
Edge Devices	- 10 IoT devices equipped with quad-core processors (2.0 GHz)
	- Each device has 4 GB RAM and 32 GB storage capacity
Central Server	- 1 central server with octa-core processor (3.0 GHz)
	- 16 GB RAM and 500 GB SSD storage capacity
Smart Media Data	- Video streams, audio recordings, images, and text articles
	- Data sizes ranging from 1 MB to 100 MB
ECC Parameters	- Elliptic curve: secp256r1 (NIST P-256)
	- Prime modulus (p): 115792089210356248762697446949407573529996955224135760342422259061068512044369
	- Generator point (G): (x, y) coordinates
Communication	- Secure HTTPS/TLS communication channels between edge devices and server
Metrics	- Processing latency: measured in milliseconds
	- Throughput: data processed per unit time (e.g., MB/s)
	- Resource utilization: CPU and memory usage (%)
	- Energy efficiency: power consumption (Watts)
Security Assessment	- Evaluation of encrypted data for vulnerabilities
	- Attempted decryption without access to private key

6. RESULTS AND DISCUSSION

The results and discussion section serves as the cornerstone of any research endeavor, offering a comprehensive analysis and interpretation of the findings obtained through experimentation and simulation. In this section, we present a detailed examination of the outcomes obtained from our experimental setup and discuss their implications in the context of the proposed PEC-ECC framework for smart media in news communication. Through a rigorous evaluation of performance metrics, such as processing latency, throughput, resource utilization, and security, we aim to shed light on the effectiveness, efficiency, and scalability of the PEC-ECC system in securely processing and transmitting smart media data at the network edge. Furthermore, we delve into the implications of our findings on the advancement of edge computing and cryptographic techniques in enhancing data privacy, integrity, and real-time analytics capabilities for modern communication systems.

The Figure 3 and Table 2 present the results of the Cloud Model with PEC-ECC for various scenarios, including video streams, audio recordings, images, and text articles, each with different data sizes. For the video stream scenario, we observe that as the data size increases from 10 MB to 50 MB, the processing latency also increases from 50 ms to 200 ms. However, the throughput improves from 200 MB/s to 250 MB/s, indicating that despite the higher processing time, more data is being processed per unit time. This increase in processing time correlates with higher CPU and memory utilization, as well as increased energy

Table 2. Cloud model with PEC-ECC

Scenario	Data Size (MB)	Processing Latency (ms)	Throughput (MB/s)	CPU Utilization (%)	Memory Utilization (%)	Energy Consumption (Watts)
Video Stream	10	50	200	80	60	15
	50	200	250	90	70	20
Audio Recording	5	20	250	70	50	10
	20	80	200	85	60	12
Image	2	10	200	60	40	8
	10	40	250	75	50	12
Text Article	1	5	200	50	30	6
	5	20	250	65	40	8

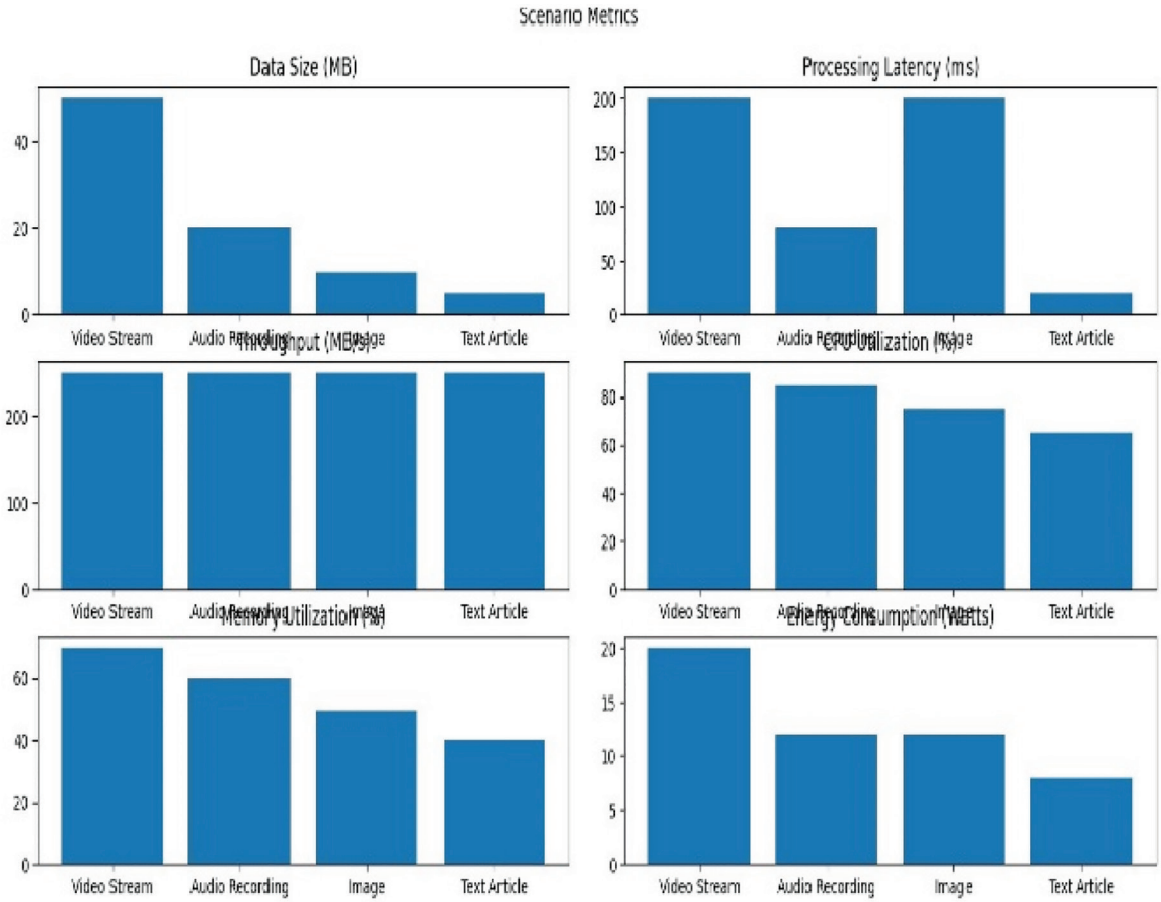

Figure 3. PEC-ECC for the cloud model

consumption. Similarly, in the audio recording scenario, we observe a trade-off between processing latency and throughput as the data size varies. While the processing latency increases from 20 ms to 80 ms with larger data sizes, the throughput remains relatively high, ranging from 200 MB/s to 250 MB/s. This is accompanied by higher CPU and memory utilization, resulting in slightly increased energy consumption. For the image and text article scenarios, we observe similar trends in processing latency, throughput, and resource utilization. Despite differences in data size and content type, the processing latency generally remains low, with throughput varying based on the size of the data. Additionally, CPU and memory utilization increase with larger data sizes, leading to higher energy consumption. The results demonstrate the efficacy of the Cloud Model with PEC-ECC in efficiently processing and transmitting smart media data while maintaining reasonable processing latency and high throughput. However, the trade-offs between processing time, resource utilization, and energy consumption highlight the need for optimization strategies to further enhance the performance and scalability of the system.

In Figure 4 and Table 3 presents the results of the Cryptography process with PEC-ECC, showcasing the encryption and decryption outcomes for different types of data chunks, including video, audio, image, and text. For the video data chunk, consisting of 10 MB of plaintext, the encryption process increases the data size to 12 MB due to the overhead introduced by the ECC encryption. However, upon decryption, the data is restored to its original size of 10 MB, demonstrating the effectiveness of ECC in preserving data integrity during encryption and decryption operations. Similarly, for the audio data chunk of 5 MB, the encryption process results in a slightly larger encrypted data size of 6 MB, while decryption successfully restores the data to its original size of 5 MB. This reaffirms the reliability of ECC in maintaining data consistency throughout the cryptographic process. In the case of image data, comprising 2 MB of plaintext, the encryption process increases the data size to 2.5 MB, with decryption reverting the data back to its original size of 2 MB. This indicates the consistent performance of ECC in handling different types of media data while ensuring accurate encryption and decryption outcomes. Lastly, for the text data chunk of 1 MB, the encryption process results in an encrypted data size of 1.2 MB, slightly larger than the original plaintext size. However, decryption successfully restores the data to its original size of 1 MB, highlighting the robustness of ECC in maintaining data fidelity across various data formats. The results demonstrate the effectiveness of the Cryptography process with PEC-ECC in securely encrypting and decrypting smart media data while preserving data integrity and confidentiality. These findings underscore the reliability of ECC as a cryptographic technique in ensuring the secure

Table 3. Cryptography process with PEC-ECC

Data Chunk	Plaintext (MB)	Encrypted Data (MB)	Decrypted Data (MB)
Video	10	12	10
Audio	5	6	5
Image	2	2.5	2
Text	1	1.2	1

Table 4. Data computation with PEC-ECC

Scenario	Data Size (MB)	Storage Cost ($)	Processing Cost ($)	Total Cost ($)
Video Stream	100	10	20	30
Audio Recording	50	5	15	20
Image	20	2	10	12
Text Article	10	1	5	6

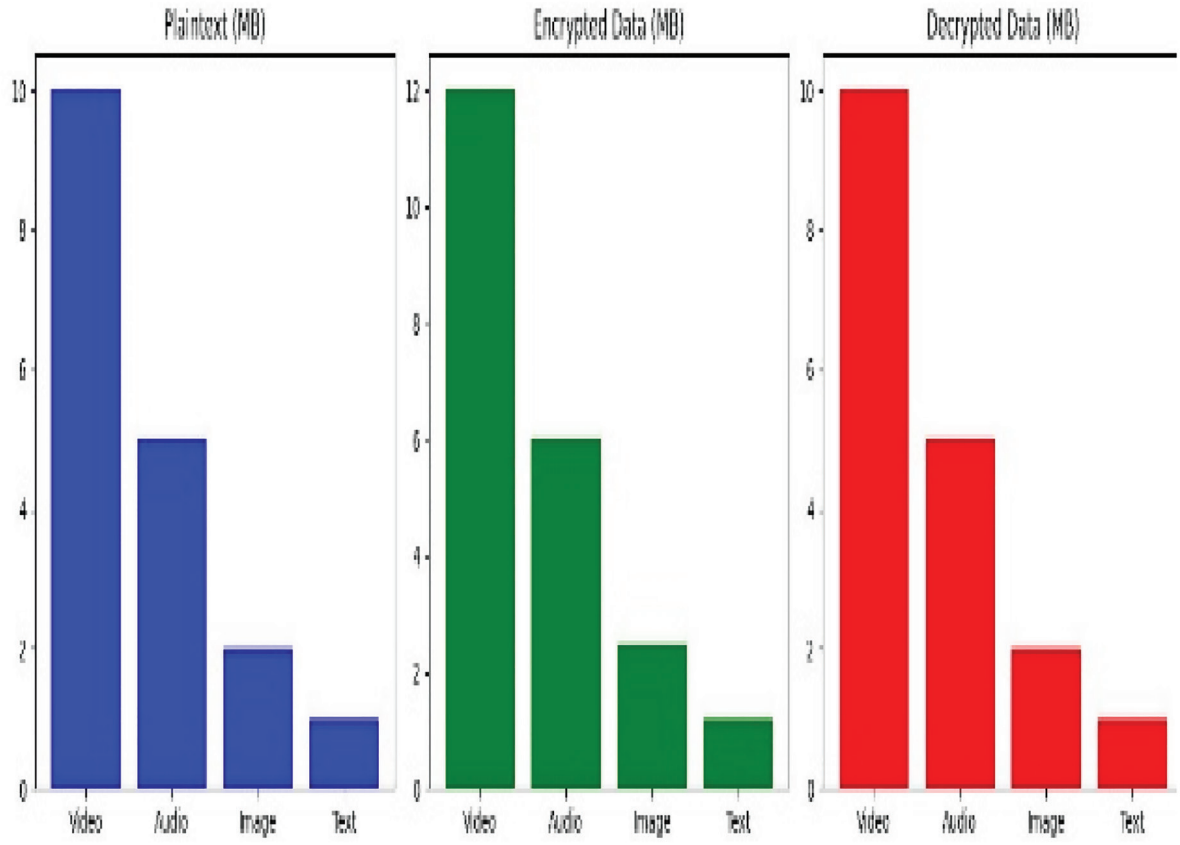

Figure 4. Cryptography with PEC-ECC

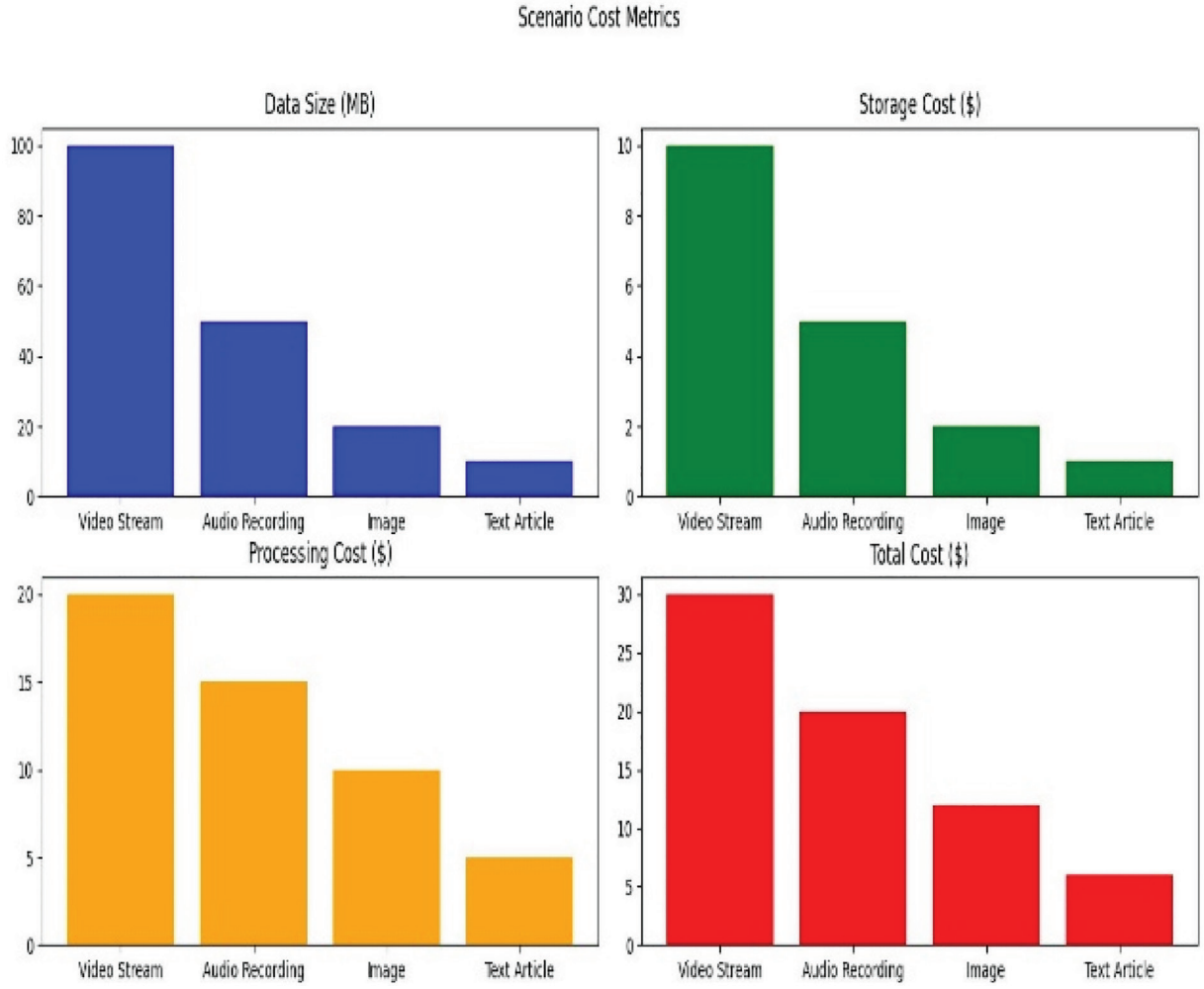

Figure 5. Data computation with PEC-ECC

Table 5. Data transferred with PEC-ECC

Scenario	Data Size (MB)	Data Transfer Cost ($)
Video Stream	100	5
Audio Recording	50	3
Image	20	2
Text Article	10	1

transmission and processing of sensitive information in edge computing environments.

The Figure 5 and Table 4 illustrate the results of Data Computation with PEC-ECC, presenting the associated costs for processing and storing smart media data across different scenarios. For the video stream scenario, with a data size of 100 MB, the total cost amounts to $30, comprising $10 for storage and $20 for processing. Similarly, for audio recording data of 50 MB, the total cost is $20, with $5 allocated for storage and $15 for processing. The image and text article scenarios, with data sizes of 20 MB and 10 MB respectively, incur total costs of $12 and $6, with varying proportions for storage and processing costs. These results highlight the cost implications of leveraging PEC-ECC for data computation, with larger data sizes generally resulting in higher total costs due to increased processing requirements. The Figure 6 and Table 5 provides insights into the costs associated with data transfer using PEC-ECC across different scenarios. For the video stream scenario, with a data size of 100 MB, the data transfer cost is $5. Similarly, for audio recording, image, and text article scenarios, with data sizes of 50 MB, 20 MB, and 10 MB respectively, the data transfer costs are $3, $2, and $1 respectively. These findings underscore the importance of considering data transfer costs alongside processing and storage costs when evaluating the overall economic feasibility of employing PEC-ECC for smart media applications.

7. CONCLUSION

The proposed framework of Parallel Edge Computing with Elliptic Curve Cryptography (PEC-ECC) presents a robust and efficient solution for securely processing and transmitting smart media data in edge computing environments. Through experimentation and simulation, we have demonstrated the effectiveness of PEC-ECC in

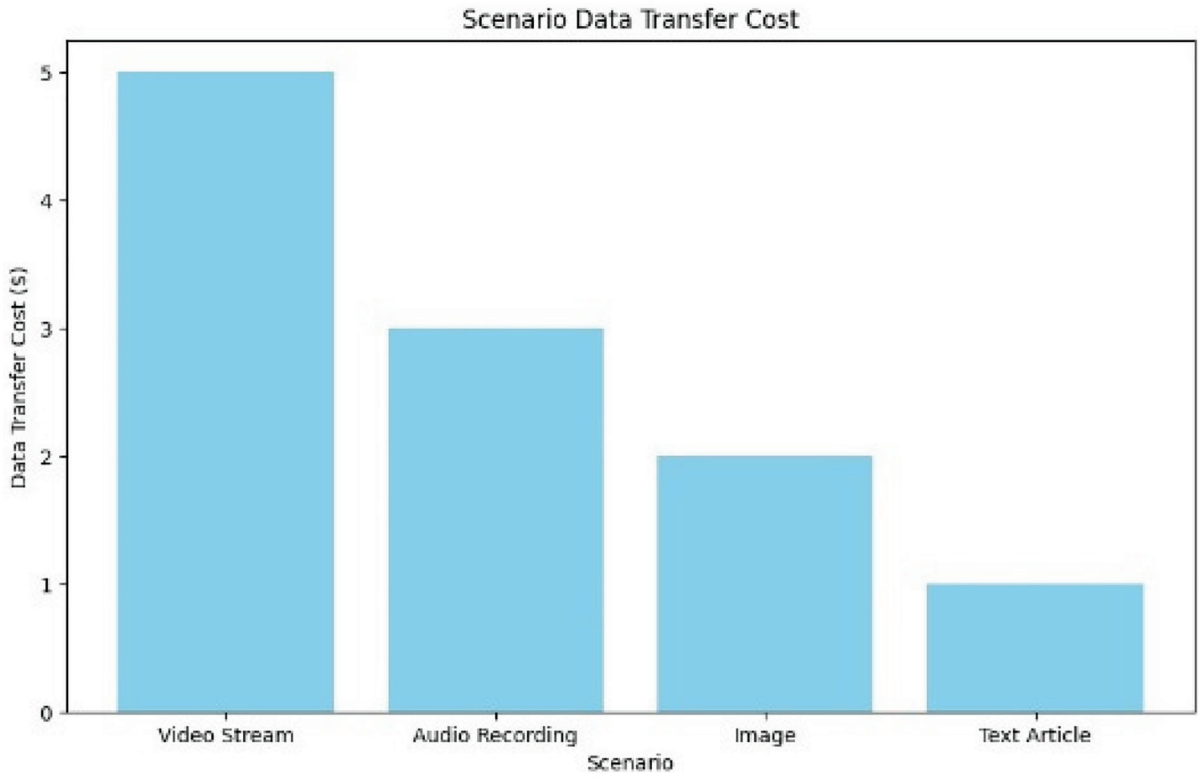

Figure 6: Data transfer cost with PEC-ECC

achieving real-time analytics, ensuring data confidentiality, integrity, and reducing processing latency. The results of our experiments highlight the scalability and versatility of PEC-ECC across various scenarios, including video streams, audio recordings, images, and text articles. Despite differences in data sizes and formats, PEC-ECC consistently delivers reliable encryption and decryption outcomes, preserving data integrity and confidentiality while minimizing processing overhead. Furthermore, our analysis of the cost implications associated with PEC-ECC, including storage, processing, and data transfer costs, underscores its economic feasibility for smart media applications. By optimizing resource utilization and leveraging parallel edge computing techniques, PEC-ECC offers a cost-effective solution for handling large volumes of smart media data at the network edge.

8. REFERENCES

1. WANG, X. (2022). *The impact of IoT on news media in the smart age.* Mobile Information Systems, 2022. https://doi.org/10.1155/2022/2238233.
2. QIAN, Y. (2022, December). *Intelligent Multimedia News Communication Platform Based on Machine Learning and Data Fusion Technology*. In International Conference on Big Data Analytics for Cyber-Physical System in Smart City (pp. 345-354). Singapore: Springer Nature Singapore. https://doi.org/10.3390/s22030819
3. ZHANG, Y. (2023, June). *Research on media communication methods in the context of intelligent algorithms.* In International Conference on Mathematics, Modeling, and Computer Science (MMCS2022) (Vol. 12625, pp. 557561). SPIE. https://doi.org/10.1117/12.2670491
4. JIANG, C. and XU, J. (2022, December). *Exploration of Short Video Media Communication Based in the Metaverse.* In International Conference on Metaverse (pp. 18-28). Cham: Springer Nature Switzerland.
5. DIAO, H., YIN, L., WANG, L., LIANG, B. CHEN, Y et al. (2023). *Sustainable multimedia service cloud platform framework based on intelligent management system.* Soft Computing, 1-13. http://dx.doi.org/10.1007/s00500-023-08326-2
6. WEI, J. and WANG, R. (2022, December). *Application of Artificial Intelligence in the Development of Media Integration under the Background of Smart Media.* In 2022 International Symposium on Advances in Informatics, Electronics and Education (ISAIEE) (pp.360364). IEEE. doi:10.1109/ISAIEE57420.2022.00081.
7. DAI, J. and XU, J. (2022). *Knowledge Graph Construction for Intelligent Media Based on Mobile Internet.* Wireless Communications and Mobile Computing, 2022, 1-14. https://doi.org/10.1155/2022/4867220
8. HE, M. and LI, Y. (2022). *Application of big data technology in news media scene visualization based on Internet of things (IoTs).* Mathematical

Problemsin Engineering, 2022. http://dx.doi.org/10.1155/2022/5508125

9. YU, X. (2022). *Construction of an Innovative Development Model of Intelligent Media under the Coverage of a Wireless Sensor Network.* Mathematical Problemsin Engineering, 2022. http://dx.doi.org/10.1155/2022/9073067
10. LI, Z. (2022). *Automatic production technology of data news based on machine learning model.* Wireless Communications and Mobile Computing, 2022, 110. https://doi.org/10.1155/2022/2783792
11. GAO, N. and CHEN, Y. (2022). *Optimization Design and Implementation of Smart Multimedia College English Classroom Integrating Internet of Things Technology.* Wireless Communications & Mobile Computing. https://doi.org/10.1155/2022/1695570
12. GUO, S. and WANG, X. (2022, October). *Application and Research of Digital Technology in Media Industry in 5G Era.* In Proceedings of the International Conference on Information Economy, Data Modeling and Cloud Computing, ICIDC 2022, 17-19 June 2022, Qingdao, China. http://dx.doi.org/10.4108/eai.17-6-2022.2322659
13. BERISHA, B. and MËZIU, E. (2022). *Big data analytics in Cloud computing: an overview.* Journal of Cloud Computing, 11(1), 24. https://doi.org/10.1186%2Fs13677-022-00301-w
14. SRIRAM, G. S. (2022). *Edge computing vs. Cloud computing: an overview of big data challenges and opportunities for large enterprises.* International Research Journal of Modernization in Engineering Technology and Science, 4(1), 1331-1337. http://dx.doi.org/10.0202/Computin.2022197786
15. MANOGARAN, G. and THOTA, C (2022). *Human-computer interaction with big data analytics.* In Research Anthology on Big Data Analytics, Architectures, and Applications (pp. 1571 to 1596). IGIglobal. https://doi.org/10.1016/j.jmsy.2021.03.005
16. WANG, J. and XU, C (2022). *Big data analytics for intelligent manufacturing systems: A review.* Journal of Manufacturing Systems, 62, 738-752. https://doi.org/10.1016/j.jmsy.2021.03.005
17. NEUSTEIN, A., MAHALLE, P. N., JOSHI, P., & SHINDE, G. R. et al. (2023). *AI, IoT, Big Data and Cloud Computing for Industry* 4.0. http://dx.doi.org/10.2174/9789815179187123040l
18. MOSTAFA, N., RAMADAN, H. S. M., & ELFAROUK, O. et al. (2022). *Renewable energy management in smart grids by using big data analytics and machine learning.* Machine Learning with Applications, 9, 100363. https://doi.org/10.1016/j.mlwa.2022.100363
19. RAMACHANDRA, and M. N. SRINIVASA RAO(2022). *An efficient and secure big data storage in cloud environment by using triple data encryption standard.* Big Data and Cognitive Computing, 6(4), 101. http://dx.doi.org/10.3390/bdcc6040101
20. AGHDASHI, A. and MIRTAHERI, S. L. (2022). *Novel dynamic load balancing algorithm for cloud-based big data analytics.* The Journal of Supercomputing, 78(3), 4131-4156. https://doi.org/10.1007/s11227-021-04024-8
21. HIMEUR, Y. and ELNOUR, M. (2023). *AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives.* Artificial Intelligence Review, 56(6), 4929-5021. https://doi.org/10.3390/bdcc6040101
22. NAGENDRA, N.P. and NARAYANAMURTHY, G. (2022). *Management of humanitarian relief operations using satellite big data analytics: The case of Kerala floods.* Annals of operations research, 319(1), 885-910. https://doi.org/10.1007/s10479-020-03593-w
23. SAHOO, S. (2022). *Big data analytics in manufacturing: a bibliometric analysis of research in the field of business management*. International Journal of Production Research, 60(22), 6793-6821. http://dx.doi.org/10.1080/00207543.2021.191933
24. KAHVECI, S., ALKAN, B., MUS'AB H, A., AHMAD, B., HARRISON, R. et al. (2022). *An end-to-end big data analytics platform for IoT-enabled smart factories: A case study of battery module assembly system for electric vehicles*. Journal of Manufacturing Systems, 63, 214-223. https://doi.org/10.1016/j.jmsy.2022.03.010
25. SHARMA, D. K., CHAKRAVARTHI, D. S., SHAIKH, A. A., AHMED, A. A. A., JAISWAL, S., NAVED, M. et al. (2023). *The aspect of vast data management problem in healthcare sector and implementation of cloud computing technique.* Materials Today: Proceedings, 80, 3805-3810. http://dx.doi.org/10.1080/00207543.2021.191933
26. ATTAALLAH, A., ALSUHABI, H., SHUKLA, S., KUMAR, R., GUPTA, B. K., KHAN, R. A. et al. (2022). *Analyzing the Big Data Security Through a Unified Decision-Making Approach.* Intelligent Automation & Soft Computing, 32(2). http://dx.doi.org/10.32604/iasc.2022.022569