Edge-Aware Offloading for Minimizing Bandwidth Costs in Edge Nodes-Cloud Servers Architecture Using Secure Measures

Buchanagandi E. Nyamajeje * , Lawrence Kerefu , Huiqun Yu

Posted Date: 4 March 2026

doi: 10.20944/preprints202603.0270.v1

Keywords: cloud computing on the mobile; reducing bandwidth; secure multi-party computation; adaptive data management; task offloading; game theory; edge computing

 doi:10.20944/preprints202603.0270.v1

Article

Edge-Aware Offloading for Minimizing Bandwidth Costs in Edge Nodes-Cloud Servers Architecture Using Secure Measures

Buchanagandi E. Nyamajeje [1,*], Lawrence Kerefu [1] and Huiqun Yu [2]

[1] Department of Digital Technologies and Information Science, Dar es Salaam Tumaini University, Tanzania
[2] Department of Computer Science and Engineering, East China University of Science and Technology, China
* Correspondence: buchanagandi.nyamajeje@dartu.ac.tz

Abstract

Mobile-based cloud computing (MBCC) has become a paradigm shift that greatly enhances the computing power of the mobile devices with limited resources by enabling them to access in-the-cloud resources of highly scalable infrastructures at considerable distance. Nevertheless, the constant and unregulated data transfer between mobile customers and remote cloud providers is deep bandwidth expenses, and at the same time, interfere with the overall system performance, address undesirable energy utilization, and sensitive information to security weaknesses. These are especially the problem in bandwidth-limited geographical locations or cost-aware enterprise settings in which the network usage is directly translated into expensive operational costs. This paper proposes a hierarchical, edge-aware architecture of the system in an approach to provisioning a holistic optimization of bandwidth usage as long as end-users remain robust in performance, in no less than three-way integration of mobile devices, localized, edge, and centralized cloud data centers. In this proposed ecosystem tasks to be offloaded are strictly encrypted and coded before their offloading and preserving sensitive user information as well as mathematically minimizing the encoded payload to the minimum. Also, we introduce a new hybrid approach that combines the data compression algorithms and game-theory-based optimization of tasks offloading, dynamic synchronization processes, and protocols of secure transmission through the encrypted and encoded task packing. Finally, this study can be used in regards to the basic development of more sustainable, secure, and cost-efficient mobile-cloud systems that are highly essential in the next generation of Internet of Things (IoT) applications in cost-sensitive environments.

Keywords: cloud computing on the mobile; reducing bandwidth; secure multi-party computation; adaptive data management; task offloading; game theory; edge computing

1. Introduction

1.1. Background

Mobile computing is a field which is undergoing an unprecedented period of massive growth in the modern era of digital connectivity everywhere. A continuously expanding smart device ecosystem - smartphones, smart wearable sensors, intelligent notebooks, smart autonomous IoT devices, and more - has been spread exponentially around the world. These new devices have inherent new data-creating resources, including high-resolution cameras, environmental sensors and sophisticated user interface modules. On the one hand, these features allow effective online interaction, multimedia stream-based immersion and intricate gaming, but on the other hand, large amounts of raw data are being generated, which demand significant levels of processing compute. Mobile-based Cloud Computing (MBCC) was developed with an idea of combating the insufficiency of mobile gadgets to provide any significant number of computational power and the excessive

 doi:10.20944/preprints202603.0270.v1

requirements of the latest applications [3]. Mobile devices can save power, minimize thermal throttling and take advantage of the virtually infinite resources of centralized data centers by offloading process-intensive tasks to remote cloud servers. Nonetheless, the classic offloading cloud model is cloud-centric in nature; essentially, the model heavily depends on the transfer of large data sets via wide-area networks (WANs) and this creates prohibitive latencies and tap-locks core network backbones. In order to avoid this physical constraint, the Mobile Edge Computing (MEC) paradigm has become wide used [31] MEC decentralizes the cloud computing and storage by using computing and storage capabilities at the periphery of the network- usually alongside cellular base stations or Wi-fi access points [25]. This physical proximity enables ultra-low latency changing of process, contextual knowing and real-time decisions and has transformed the way applications will process compute-intensive workloads.

1.2. Motivation

Although MEC has a variety of architectural benefits, the inherent issue of overhead control in data transmission has been a major bottleneck. Since the sheer amount of data passing through the network keeps getting higher, the unrelenting transmission of data between the mobile clients, edge servers, and the central cloud are costly in terms of bandwidth. Network bandwidth is a limited, purely rationed and extremely costly good in most practical deployment locations, including remote industrial locations, smart agriculture or emerging markets. Despite using methods of offloading that are as old as data transmission, most traditional offloading algorithms have been largely preoccupied with the goal of minimizing task execution latency or minimizing device-side energy consumption, often with the false belief that network bandwidth is effectively free and indefinitely limitless [2] As very high rates of devices simultaneously start to offload to the cloud or the edge without a coordinated bandwidth-sensitive approach, the network congestion results into packet loss, extreme transmission delays, and overwhelming financial expenses to both service providers and end-users. Thus, there is an urgent need to transform into a green and economical mobile cloud computing model. This shift demands the creation of advanced, bandwidth-focused mathematical modeling so that network capacity is not only considered as a physical bottleneck, but also one of the key variables of financial costs that needs to be systematically reduced [7].

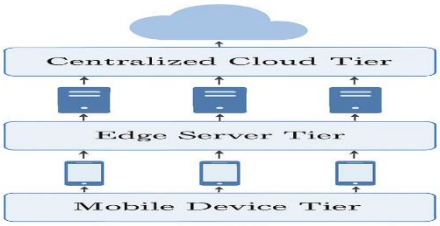

Figure 1. This layered view depict how computation and storage responsibilities are distributed across the hierarchy and controls for upward flow of data and processing, from devices to edge or cloud.

1.3. Security Challenges In Edge-Cloud Ecosystems

At the same time, the architectural transformation of a distributed edge-cloud ecosystem significantly increases the scope of attack by an attacker. The data will have to traverse many, possibly insecure network hops, when offloading computation to a physically secure mobile device to a distributed system of edge nodes and cloud servers. This puts personal health records as well as proprietary industrial data in the hands of eavesdropping, man-in-the-middle (MitM) attacks and potentially altered data [26]. Old fashioned centralized perimeters of security are completely unsuitable when it comes to MEC. As a result of that the strong secure systems like Advanced Encryption Standard (AES) cryptography and secure task packing will need to be incorporated directly in the offloading pipeline [21]. Nevertheless, the cost of cryptographic computation and tag

 doi:10.20944/preprints202603.0270.v1

appendage on authenticated data is not a trivial cryptography computation and data expansion, which counterproductively adds to not only execution latency, but also bandwidth utilization. Striking the balance between the necessity of strict security and the intentions to reduce the bandwidth expenses, is a multi-objective, highly complicated optimization problem which is the main centre of this study.

1.4. Research Contributions

In order to thoroughly tackle both overlapping issues of the optimization of bandwidth costs and data security in mobile cloud computing, this paper suggests the systematic edge-aware offloading framework. The major findings of this research can be described as follows:

1. Hierarchical System Architecture: We develop a sound and strong three-level architecture that includes mobile machines, peripheral servers, and centralized cloud servers, which support the allocation of tasks dynamically and granularly according to the network situation presented in real time.

2.Bandwidth-Centric Cost Model: We create a strict mathematical optimization created that explicitly costs data transmission with the consumption of uplink and downlink bandwidth forms the main objective function to be minimized.

3. Integrated Security Overhead Modelling: we present an innovative security layer, which represents in mathematical terms the data enlargement and the computation cost due to cryptographic encryption, and computation due to encryption, while guaranteeing that security considerations can be included in the offloading decision list.

4. Hybrid Game-Theoretic Technique: We introduce a novel hybrid algorithm enabling an adaptive data compression along with game-theory-based decision-making, which allows the distributed, decentralized task offloading and provides practically optimal bandwidth minimization.

5. Empirical validation: We empirically prove by means of forums in which we extensively simulate a prototype framework with real-world application profiles that the proposed technique will lower the total bandwidth expenses by up to 40 percent without compromising hard bandwidth limits.

2. Literature Review

Mobile cloud computing and edge offloading landscape has been extensively experimented within the last ten years. This section summarizes the technological breakthroughs in offloading, bandwidth optimization and safe edge computing that are crucial and unveils areas in research where existing gaps exist and help close them with the study.

2.1. Computation Offloading in Mobile Edge Computing

General linking Computation offloading can be viewed as the algorithm of distributing computationally intensive tasks inside a mobile device to an external platform with more resources at its disposal. Initial studies within this field were more specifically concerned with binary offloading to centralized clouds, with a task being performed on a single platform entirely, or being completely sent to the cloud. Nevertheless, as mobile applications grew in sophistication, partial offloading, in which applications were cut into modular units, became popular. Latency-conscious, multi-server partial task offloading has been explored in much detail in recent literature to balance loads and minimize execution delays in edge computing environments [28]. Game theory has become a popular mathematical instrument in order to cope with the complexity of distributed decision-making in multi-user environments. Decentralized offloading Game-theoretic strategies permit many devices (as independent agents) to compete over scarce edge resources. An example case is that recent literatures focus on utilizing Stackelberg games to model the interaction between an Edge Server Provider (ESP), who becomes a leader and sets computing resource prices, and mobile users who become followers and calculate their optimum offloading strategies depending on prices [9]. On the

 doi:10.20944/preprints202603.0270.v1

same note, bi-level distributed computation offloading algorithms are developed to achieve a Nash equilibrium, which will maximize the profit of device managers and make the allocation of resources across edge nodes equitable [16]. Although these game-theoretic models are effective in controlling the allocation of computing resources and prices, they often do not offer an explicit financial signal of the actual cost of the network bandwidth that is necessary to ferry the data to the edge servers.

2.2. Bandwidth and Energy Optimization

The energy consumption and bandwidth allocation are at an intersection and this has led to the design of different optimization heuristics. Conventional heuristic methods, including the Ant Colony Optimization (ACO) and Genetic Algorithms (GA), have also been implemented in hybrid queue-based algorithm lays to control task allocation in two-tier mobile cloud systems including the localized cloudlets and the public clouds [11]. These algorithms are able to model the queuing delays at the edge nodes and attempt to balance the load in order to avoid overloading the servers.

Yet, to the extent that the literature on optimization has gone to date, bandwidth is viewed as being merely a constraint (i.e. maximum achievable throughput) and not an actual operational cost. In situations where the transmission of data is charged like in cell 5G networks or commercial satellite connections, it is urgent to ensure that the raw number of bytes transmitted is minimal. The recent proposals have tried to curb the bandwidth consumption with the help of data reduction mechanisms, yet they hardly combine the mechanisms with the dynamism of task scheduling [29]. Our study will fill this gap by developing directly the financial cost of bandwidth as the main minimization goal, using data compression and task packing as active constituents of the offloading decision.

2.3. Secure Offloading in Edge-Cloud Systems

Security is also one of the top priority concerns that largely determine viability of outsourcing computation. Multi-tenant and distributed characteristics of MEC predispose it to breaches of data [12]. Recent developments have suggested the application of deep frame training with strong security functions, including dynamic one-time encrypted AES information, to safeguard the data transmission during offloading calculation [5]. Moreover, the models of an energy-conscious and secure offloading of tasks have been developed in the form of mixed-integer programming problems, aiming to minimize the energy of the system and still keeping its latency constant and securing it with the help of required cryptographic means [2].

With all of these improvements, the serious error in the literature is underestimation of the security overhead. The encryption algorithms and integrity checking naturally add to the payload size (block padding and initializing vectors) and require more CPU processor cycles to have an encryption/decryption procedure [27] Even the current frameworks consider security to be an abstract constraint and indiscriminately fail to mathematically penalize the offloading algorithm due to the high bandwidth cost caused by the expansion of secure data. We explicitly model this data bloat generated by security, as the offloaded tasks are safely encrypted and coded, though at the same time we do not allow the overall bandwidth consumption to be anything other than highly optimized.

2.4. Research Gap Analysis

We find that there are no less than critical gaps in the current literature:

- Absence of Explicit Bandwidth Cost Modelling: The majority of studies assume bandwidth to be seen as a physical constraint, not as a financial cost variable that is to be minimized.
- Lack of Security Overhead Quantification: Security is applied in an abstract way without mathematical modelling on its effect on the size of the data payload and computational costs.
- Lack of Integrated Hybrid Solutions: There are only a few structures that integrate the three concepts of data compression, secure encryption, and optimization of a game-theoretic model of decision making.

 doi:10.20944/preprints202603.0270.v1

- Lack of Real-World testing: A large number of suggested algorithms have not been thoroughly tested or validated in a testbed or in realistic network conditions with variable density of devices that vary with time.

The current research directly closes these gaps by the proposal of the Edge-Aware Secure Offloading (EASO) framework that fully incorporates all these dimensions.

3. System Architecture

To accomplish the best bandwidth cost minimization and assure task implementation, the present study provides a detailed hierarchical architecture, which is the Mobile Device Tier: smartphone, tablets, IoT sensors, followed by the Edge Server Tier: gateways, edge servers, micro data centers, and the Centralized Cloud Tier: larger-scale cluster and data centers as illustrated in Figure 2(a). In this approach we have considered **Mobile Layer**: Generates task T_i, **Edge Layer (MEC)**: Low-latency intermediate computing, **Cloud Layer**: High computational capacity, and **Decision Engine**: Minimizes bandwidth cost and ensures latency constraint satisfaction, Forward path for task offloading and Return path for output data delivery for computational and storage responsibilities as shown in Figure 2(b).

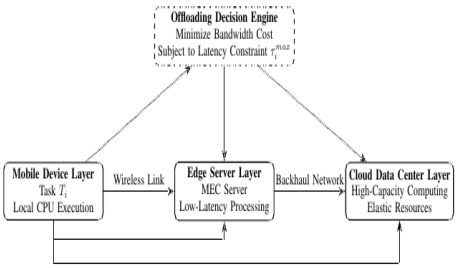

Figure 2(a): System level Mobile-Edge-Cloud offloading Architecture

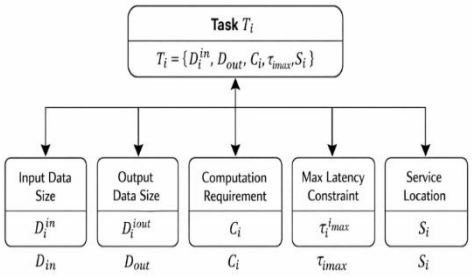

Figure 2(b). Computation task model components

Figure 2.

Each layer and its corresponding functionality is elucidated below:

3.1. Mobile Device Tier (Data Generation):

Its bottom level includes heterogeneous mobile and IoT devices, such as smartphones and wearable sensors, as well as autonomous cars. These machines are the origin and major data producers with respect to computationally intensive operations [10]. Every device has localized, though limited, computational information, storage and battery control. At the creation of every application, a localized decision engine inside the device considers the properties of the task, e.g., the raw input size, needed CPU cycles, latency deadline, and sensitivity of the security. According to these parameters, the device should resolve up to which state is the task to be executed or it should make the secure offloading process. In case of offloading, the device uses adaptive data compression and cryptographic encoding to the payload in order to reduce the transmission footprint and secure confidentiality of the data prior to its transmission by the wireless uplink [30].

 doi:10.20944/preprints202603.0270.v1

3.2. Edge Server Tier (Proximity Processing)

The intermediate level is composed of edge servers (or cloudlets), which are strategically positioned physically close to the mobile devices and are often directly co-located with 5G cellular radio base stations or enterprise access points [18]. These edge nodes are highly computerized in contrast to mobile devices but have limited capacity as compared to the centralized cloud. The major purpose of the Edge Tier is to query offloaded tasks, initial security decoding, and run directly on the Edge Tier with the latency-sensitive applications with a minimum delay on the network [6]. Also, edge servers can be viewed as smart routing gateways, so once an edge node falls into peak utilisation or a particular task demands an astronomical number of simultaneous processors, which the edge cannot provide, the compressed and encrypted task can be dynamically re-routed to the central cloud [8]. This layer plays a very important role in preserving real time adaptive synchronization and avoiding local network congestion.

3.3. Centralized Cloud Tier (High-Capacity Processing)

The top level is centralized, large-scale cloud data centres (e.g. AWS, Microsoft Azure). The tier is associated with virtually limitless computational, storage and processing capabilities. It is best applicable to high-computation but low-latency tasks. But the transmission of data via a mobile device, through an edge node, and into the centralized cloud is the most expensive in terms of transmission delay and maximum bandwidth since it requires WAN backbones to be metered [15]. The suggested hybrid method carefully controls the routing of activities to this tier, and it is only used when the computational benefits exceed the hefty financial bandwidth fines as well as the security latency costs.

3.4. Security and Communication Framework

A solid security and communication are what holds these three levels together. The links between the devices and the edge (through 5G/Wi-Fi) as well as the edge and cloud (through optical fibre or broadband) are all models as untrusted networks by default. In order to fight this, a Secure Multi-Party Computation (SMPC) informed strategy is incorporated [1]. Lossless algorithms are used to ruthlessly compress data in order to remove redundant bits. The AES-based encryption of the compressed payload is then followed by its encapsulation into normal task packets. This is to guarantee that even in the event that the packets are intercepted either on the WAN or on the radio access network, the proprietary data cannot be read in unencrypted form, and is inaccessible and unaltered.

4. Problem Formulation

In order to mathematically optimize the offloading process, we develop a strict mathematical model that measures bandwidth cost, latency, energy, and security overheads.

4.1. System Model Definitions

Let $\mathcal{N}$ N= {1, 2 …N} denote a set of mobile devices, and $\mathcal{M} = \{1,2,\dots,M\}$ denote a set of available edge servers. The centralized cloud is denoted as node C.For any mobile device $i \in \mathcal{N}$, an application generates a task T_i defined by a tuple $T_i \triangleq \{D_i^{in}, D_i^{out}, C_i, \tau_i^{\max} S_i\}$ where:

- D_i^{in} : The raw input data size (in bits) required to process the task.
- D_i^{out} : The expected output data size (in bits) generated after execution.
- C_i : The computational workload required to execute the task (in total CPU cycles).
- $\tau_i^{\max}$: The strict maximum tolerable delay (deadline) for task completion (in seconds)
- S_i : The binary security requirement flag (1 if encryption is mandated, 0 otherwise).

 doi:10.20944/preprints202603.0270.v1

- The offloading decision for device i is governed by a variable $x_i \in \{0,1,2\}$, where:
- $x_i = 0$: Task executes locally on the device.
- $x_i = 1$: Task is offloaded to the nearest edge server $m \in \mathcal{M}$.
- $x_i = 2$: Task is offloaded to the centralized cloud C.

4.2. Security Overhead and Data Compression Model

To minimize bandwidth, device i applies a compression algorithm yielding a compression ratio γ (0<γ≤1). Following compression, if $S_i = 1$, AES encryption is applied. Encryption inherently adds a data expansion factor α (due to padding, initialization vectors, and MAC tags) where $\alpha \geq 1$. Thus, the actual transmitted payload size $\tilde{D}_i^{in}$ is modelled as:

$$\tilde{D}_i^{in} = \begin{cases} \alpha \cdot (\gamma \cdot D_i^{in}) & if\ S_i = 1 \\ \gamma \cdot D_i^{in} & if\ S_i = 0 \end{cases} \quad (2)$$

Similarly, the encrypted output data returning to the device is:

$$\tilde{D}_i^{out} = \begin{cases} \alpha \cdot (\gamma \cdot D_i^{out}) & if\ S_i = 1 \\ \gamma \cdot D_i^{out} & if\ S_i = 0 \end{cases} \quad (3)$$

Additionally, encryption and decryption require computational effort. Let ω represent the CPU cycles required to encrypt/decrypt one bit of data. The total security computational overhead for task T_i is:

$$C_i^{sec} = S_i \cdot \omega \cdot (D_i^{in} + D_i^{out}) \quad (4)$$

4.3. Communication and Bandwidth Model

According to Shannon's capacity theorem, the wireless uplink transmission rate R_i^{up} between device i and the edge server is defined as:

$$R_i^{up} = B_i log_2 \left(1 + \frac{P_i \cdot h_i}{\sigma^2 + I}\right) \quad (5)$$

where B_iBi is the allocated channel bandwidth, P_i Pi is the transmission power of the device, h_ii represents the channel gain, $\sigma\sigma^2$ is the background Gaussian noise power, and I denotes the interference from other devices. The transmission delay to upload the data to the edge is:

$$t_i^{tx} = \frac{\tilde{D}_i^{in}}{R_i^{up}} \quad (6)$$

If the task is further routed to the cloud ($xi = 2$), an additional WAN transmission delay t edge →cloud is incurred, which is a function of the WAN bandwidth capacity R_{WAN}:

$$t_{i,WAN}^{tx} = \frac{\tilde{D}_i^{in}}{R_{WAN}} + t_{prop} \quad (7)$$

where t prop is the prop delay, which is the physical delay to the remote cloud center.

4.4. Bandwidth Cost Model

The crux of this research revolves around the monetization of data transfer. We define $\Phi_{edge} edge$ as the cost per megabit (Mb) for transmitting data over the edge cellular network, and Φ_{cloud}cloud as the cost per Mb for transit over the WAN to the centralized cloud. The total financial bandwidth cost Ψ_i for offloading task T_i is explicitly defined as:

$$\Psi_i(x_i) = \begin{cases} 0, & if\ x_i = 0\ (Local) \\ \Phi_{edge} \cdot (\tilde{D}_i^{in} + \tilde{D}_i^{out}), & if\ x_i = 1\ (Edge) \\ \Phi_{edge} \cdot (\tilde{D}_i^{in} + \tilde{D}_i^{out}) + \Phi_{cloud} \cdot (\tilde{D}_i^{in} + \tilde{D}_i^{out}), & if\ x_i = 2\ (Cloud) \end{cases} \quad (8)$$

 doi:10.20944/preprints202603.0270.v1

This discontinuous mapping function shows clearly that as much as cloud offloading can provide an unlimited amount of computational resources, it induces an exponentially growing financial bandwidth penalty as the combined cost of using a local and a wide-area network is additive.

4.5. Latency Model

The total execution latency L_i depends strictly on the offloading decision x_i:

Local Execution ($x_i = 0$)**:**

$$L_i(0) = \frac{C_i + C_i^{sec}}{f_i^{loc}} \quad (9)$$

where f_i^{loc} is the local CPU clock frequency of the mobile device.

Edge Execution ($\boldsymbol{x_i = 1}$)**:**

$$L_i(1) = t_i^{tx} + \frac{C_i + 2 \cdot C_i^{sec}}{f_m^{edge}} + \frac{\tilde{D}_i^{out}}{R_i^{down}} \quad (10)$$

(Note: Security overhead is doubled at the edge to account for decryption of input and subsequent encryption of output prior to return).

Cloud Execution ($x_i = 2$)**:**

$$L_i(2) = t_i^{tx} + t_{i,WAN}^{tx} + \frac{C_i + 2 \cdot C_i^{sec}}{f_C^{cloud}} + t_{WAN \to edge}^{tx} + \frac{\tilde{D}_i^{out}}{R_i^{down}} \quad (11)$$

4.6. Optimization Objective

Reducing the total bandwidth price per all the devices and having a strict focus on meeting the deadlines of the tasks and the edge server capacity constraints is our main objective. The mathematical formulation of the global optimization problem is:

$$\min_{\boldsymbol{X}} \sum_{i=1}^{N} \Psi_i(x_i) \quad (12)$$

Subject to the following constraints:

1. **Latency Constraint:** $L_i(x_i) \leq \tau_i^{max}, \forall i \in \mathcal{N}$
2. **Security Constraint:** Encrypted packing must be applied if $Si = 1$.
3. **Edge Capacity Constraint: :** $\sum_{\{i|x_i=1\}} (C_i + 2 \cdot C_i^{sec}) \leq C_m^{max}, \forall m \in \mathcal{M}$ (The total compute load assigned to an edge server cannot exceed its maximum processing capacity).
4. **Binary/Integer Constraint:** $x_i \in \{0,1,2\}, \forall i \in \mathcal{N}$

It is a non-linear programming (MINLP) with mixed-integer formulation. Due to the computational intractability (NP-hard) of achieving a globally optimal solution to large-scale mobile networks in real-time, we introduce a heuristic, game-theory-based hybrid method to solve offloading choices in a phased manner and obtain near-optimal solutions.

5. Proposed Methodology: Hybrid Game-Theoretic Offloading Algorithm

Since the process of global optimization of the multi-user task offloading in a three-tier architecture is shown to be an NP-hard problem of the Mixed-Integer Non-Linear Programming (MINLP) type, the task of centralized decision-making becomes computationally infeasible with large-scale Internet of Things (IoT) applications [24]. Hence, we suggest a heuristics based on a game theory, but decentralized and called Edge-Aware Secure Offloading (EASO). This hybrid method combines adaptive data compression, safe task packing and non-cooperative game theory to approach a Nash Equilibrium that would minimize bandwidth cost.

 doi:10.20944/preprints202603.0270.v1

5.1. Game-Theoretic Problem Formulation

We model the offloading decision process as a decentralized, non-cooperative multi-user game, denoted as

$$\mathcal{G} = \langle \mathcal{N}, \{\mathcal{S}_i\}_{i\in\mathcal{N}}, \{U_i\}_{i\in\mathcal{N}} \rangle \quad (13)$$

- **Players:** The set of mobile devices $\mathcal{N}$.

Strategy Space: Each player i selects a strategy $s_i \in \mathcal{S}_i = \{0,1,2\}$, representing local, edge, or cloud execution, respectively.

- **Utility Function:** The utility function $U_i(s_i, s_{-i})$ represents the perceived cost for device i, given its own strategy s_i and the collective strategy profile of all other devices s_{-i}.

Traditional offloading games minimize the utility that is strictly the latency or energy. Our bandwidth-based model has the major aim of minimizing the cost of bandwidth financing and identified latency misdemeanors heavily. The utility term is thus defined as:

$$U_i(s_i, s_{-i}) = \Psi_i(s_i) + \lambda \cdot max(0, L_i(s_i, s_{-i}) - \tau_i^{max}) + \kappa \cdot \eta(s_i, \mathcal{M}) \quad (14)$$

Where:

- $\Psi_i(s_i si)$ is the bandwidth cost function defined in Section 4.4.
- $L_i(s_i, si)$ is the total task latency. The latency depends on the strategies of other players ($s_{s\,-i\,i}$) because concurrent edge offloading increases queuing delay and interference.
- λ λ is a massively weighted penalty multiplier applied exclusively if the task violates its strict deadline τ_i^{max}.
- $\eta(si, M)$ is an edge-awareness congestion penalty. If the chosen edge server m is operating beyond its maximum CPU capacity C_m^{max}, the utility cost is exponentially raised by his term to discourage additional offloading to the same node, multiplied by the coefficient K.

The objective of every independent mobile device is to selfishly minimize its own utility cost U_i. The system will have reached a Nash Equilibrium (NE) in a scenario where none of the mobile devices can change course strategy s_i to achieve a lower utility cost, given that the strategies of all other devices remain unchanged.

5.2. Secure Multi-Party Computation and Task Packing

Before a task is offloaded ($s_i \in \{1,2\}$), the EASO framework enforces an "Encrypted and Encoded Task Packing" phase, inspired by Secure Multi-Party Computation (SMPC) principles.

1. **Adaptive Compression:** In order to lighten the base payload, a lossless compression algorithm (lzma) is used. The local decision engine is a part of the device and determines the data entropy. In case the data is very redundant (e.g. continuous sensor telemetry), the compression ratio γ drops significantly (approaching 0.3).
2. **Dynamic Encryption:** If the task's security flag $S_i = 1$, the device generates a dynamic, one-time symmetric AES-256 key. The payload is then compressed and encrypted and an integrity MAC (Message Authentication Code) is added afterwards. This encryption procedure makes it have a data expansion factor α (typically 1.05 to 1.10).
3. **Encapsulation:** The payload is encrypted and together with the resulting offloading metadata (CPU cycles needed, deadline) is a standardized task packet. This densified format provides that edge servers have the ability to act on the data without having to decrypt the underlying sensitive payload, and decrypt only when the task is done at the edge tier [20].

5.3. Edge-Aware Secure Offloading (Easo) Algorithm

The EASO heuristic algorithm is iterative in nature. Mobile devices analyse their task generation and recommend a starting offloading strategy at discrete time intervals. The edge servers then send out their estimated load which devices revise their strategies when they face the congestion penalty η. This algorithm has a fast convergence due to the heavy deadline penalty and congestion penalties

 doi:10.20944/preprints202603.0270.v1

which cause devices to move off of congested nodes compellingly distributing the compute load overtime without necessarily using the WAN bandwidth cost unless further needed to meet a deadline.

Table 1. Pseudocode For the Edge-Aware Secure Offloading (EASO) Algorithm.

Algorithm 1: Edge-Aware Secure Offloading (EASO) Heuristic
***Input:** Set of mobile devices N, Edge servers M, Task set T, Parameters (γ, α)*
***Output:** Optimal strategy profile* $S^* = \{s_1^*, s_2^*, \dots, s_N^*\}$
***1:** Initialize iteration counter* $k = 0$
***2:** For each device* i *in N:*
***3:** \quad Extract task parameters* (D_i^{in}, C_i, τ_i^{max}, S_i)
***4:** \quad Calculate effective payload* $\tilde{D}_i^{in}$ *based on* γ *and* α *(if* $S_i = 1$*)*
***5:** \quad Initialize strategy* $s_i^{(0)} = 0$ *(Default to Local Execution)*
***6:** End For*
***7:** While not converged AND* $k <$ *MAX_ITERATIONS do:*
***8:** \quad* $k = k + 1$
***9:** \quad Edge servers broadcast current compute load* $C_m^{current}$ *and bandwidth pricing.*
***10:** \quad For each device* i *in N (sequentially or in randomized order):*
***11:** \quad \quad Calculate Latency* $L_i(s_i)$ *for* $s_i = \{0,1,2\}$
***12:** \quad \quad Calculate Bandwidth Cost* $\Psi_i(s_i)$ *for* $s_i = \{0,1,2\}$
***13:** \quad \quad Compute Utility* $U_i(s_i)$ *for all possible strategies.*
***14:** \quad \quad Select best response:* $s_i^{(k)} = argmin\{U_i(s_i)\}$
***15:** \quad \quad If* $s_i^{(k)} \neq s_i^{(k-1)}$*:*
***16:** \quad \quad \quad Update global strategy profile* S
***17:** \quad \quad \quad Edge server updates its projected load* $C_m^{current}$
***18:** \quad \quad End If*
***19:** \quad End For*
***20:** \quad If no devices changed their strategy in iteration* k*:*
***21:** \quad \quad Nash Equilibrium reached. Break loop.*
***22:** End While*
***23:** Return final strategy profile* S^*

6. Experimental Setup and Simulation

In order to strictly validate the proposed EASO framework we designed a full up simulation environment that is a network communication dynamics versed-computation model.

6.1. Simulation Environment

A hybrid version of Network Simulator 3 (NS-3) was adopted to model the 5G and WAN telecommunications architecture and MATLAB R2023a was adopted to run the game-theoretic offloading algorithms and calculate the mathematical matrices. The virtual design architecture is an area of 1,000 by 1,000 meters. The environment will consist of a single 5G Base Station, fitted with an MEC server (Edge Tier) and a fibre-optic backhaul service and a remote data centre simulated as AWS Data Centre (Cloud Tier). Devices of mobile IoT are evenly spread in the coverage area and move randomly with a maximum waypoint velocity between 0 and 5 m/s.

6.2. Key Simulation Parameters

System performance is very compulsory to the parameterization of computation and network expenses. The default parameter of the simulation is outlined in Table 2.

 doi:10.20944/preprints202603.0270.v1

Table 2. Key Simulation Parameters for the EASO Framework Evaluation.

Parameter	*Description*	*Value / Distribution*
N	*Number of Mobile Devices*	*50 to 200*
f_i^{loc}	*Mobile Device CPU Frequency*	*1.0 GHz to 2.5 GHz*
f_m^{edge}	*Edge Server CPU Capacity*	*100 GHz*
f_c^{cloud}	*Cloud Server CPU Capacity*	*10,000 GHz*
D_i^{in}	*Task Input Data Size*	*Uniform [5,20] Megabytes*
C_i	*Task Computational Workload*	*Uniform [500,2000] Megacycles*
τ_i^{max}	*Maximum Tolerable Latency*	*Uniform [50,200] ms*
γ	*Adaptive Compression Ratio*	*0.5 to 0.75*
α	*AES-256 Security Expansion*	*1.05 (+5% overhead)*
Φ_{edge}	*Edge Bandwidth Price*	*$0.02 per MB*
Φ_{cloud}	*Cloud WAN Bandwidth Price*	*$0.10 per MB*

6.3. Baseline Algorithms for Comparison

1.In order to put the EASO algorithm performance into perspective we compared it to three known offloading paradigms that are regularly mentioned in the modern literature:

2.Local-Only Implementation: All operations are directly implemented on the mobile devices. The cost of bandwidth is no cost, but the latency and energy metrics are often devastating to intensive jobs.

3.Cloud-Only Execution: A classic MBCC solution in which all the functionality is diverted to the edge to a central cloud. This ensures that it is very fast in computation but maximum transmission delay in a WAN and enormous expenditure on bandwidth.

4.Random Offloading: Work is loaded onto the device, the edge and the cloud randomly, without mathematical optimization and it models an unmanaged, naive network.

5.Edge-Greedy Offloading: Tasks are again offloaded to the edge server either until it is full; or tasks are dropped or evicted to execute.

7. Results and Discussion

The EASO framework empirical analysis obtained deep understanding of the complex trade-offs of bandwidth spending, latency of executions, and security overhead of mobile cloud architecture.

7.1. Bandwidth Cost Minimization

The main aim of this study is to radically overpower the operating cost in the process of transferring data over the network. The total bandwidth cost of various offloading strategies on the comparative figures in Figure 1 is a pictorial indication of the scaling of the density of mobile devices in the simulation area which varies between 50 and 200.

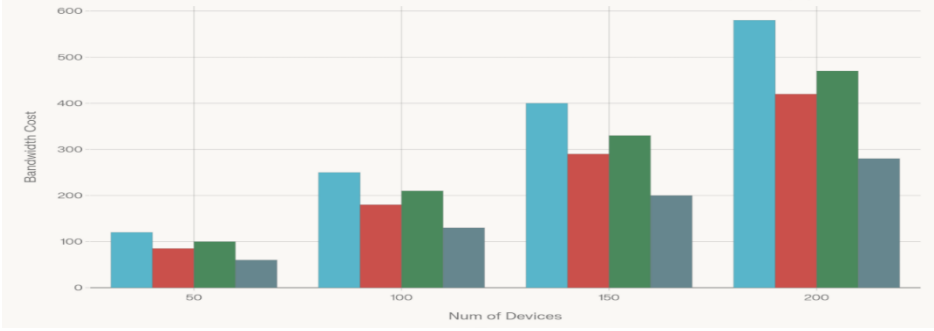

Figure 3. Comparison of Total Bandwidth Cost across different offloading strategies as device density increases.

 doi:10.20944/preprints202603.0270.v1

The Cloud-Only course of action, as shown in Figure 1, features an aggressive, linear rise in the bandwidth expenses. Since all the tasks and data created by the 200 devices have to pass through both the cellular access network and the costly wide-area network backhaul (Φ_{cloud}), the financial penalty is no longer viable on mass deployments. The Random strategy is a little better than Cloud-Only just by chance that a number of tasks are kept on the local or edge computing. On the other hand, the suggested EASO algorithm is saving bands at a dramatic cost, as much as 40 percent below Cloud-Only base, and consistently, 25-30 percent below the Edge-Greedy algorithm in high network congestion

This significant financial saving is driven by three integrated mechanisms:

1. **Task Packing:** By actively compressing the payload ($\gamma = 0.6$ average) prior to AES encryption, EASO strictly minimizes the raw bytes transmitted over the air interface.
2. **Cost-Aware Routing:** The game-theoretic utility function inherently penalizes WAN routing. EASO only selects the cloud $(x_i = 2)$ when the edge server is completely saturated, and the mobile device's local CPU cannot meet the strict deadline ($\tau_i^{\max}$).
3. **Decentralized Load Balancing:** 3. Devices can self-organize by means of Nash Equilibrium. Devices that have tasks that are highly compressible and heavy computation offload into the cloud whereas devices with large data overhead and medium computation demand remain at the edge and hence optimize the cost-per-megabit ratio.

7.2. Latency and Security Overhead Trade-Offs

When the financial cost side is absolutely vital, a framework of offloading cannot prove to be valuable unless it is capable of fulfilling the strict requirements under latency deadlines required by real-time mobile applications (e.g., autonomous driving telemetry, real-time healthcare monitoring).

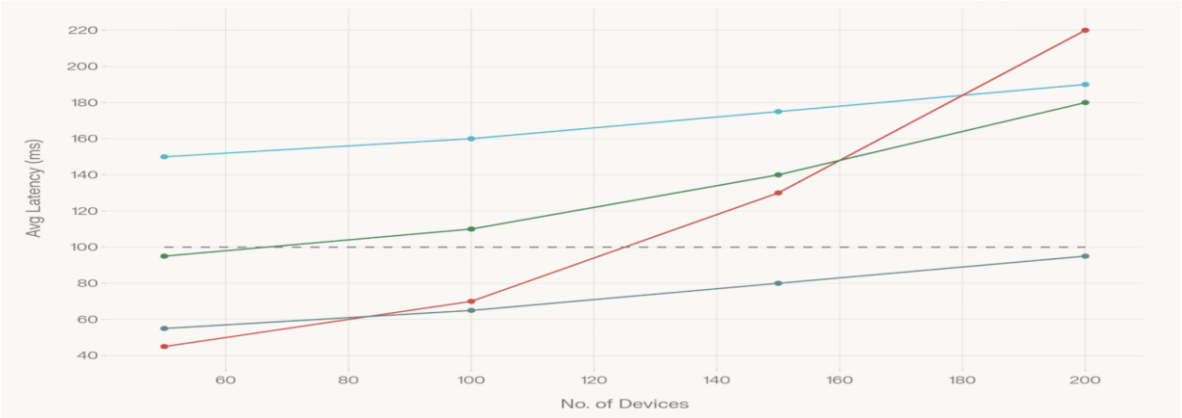

Figure 4. Average Task Execution Latency under varying mobile device densities.

Naturally added computational delay (encryption/de-encryption cycles) and transmission delay are the result of integrating security measures. Nevertheless, EASO is able to offer an average latency which is substantially below the critical value of 100 ms most of the way through the simulation profile despite this overhead.

Exceptional performance by the Edge-Greedy algorithm can be achieved only at very low densities of devices (50 devices) since at this point the edge server possesses huge CPU cycles that can be immediately utilized to execute tasks with near-zero physical distance. But with increase in the network to 150 and 200 devices, the edge server reaches its C_m^{max} capacity limit. The queuing delay in the edge node is increasing exponentially and the Edge-Greedy latency is violating the 100 ms margin and eventually even exceeding the Cloud-Only approach.

EASO wisely foresees this edge saturation. As the edge congestion penalty in the utility function takes off, EASO dynamically shifts a calculated fraction of the workload to the centralised cloud. The

 doi:10.20944/preprints202603.0270.v1

immense parallel processing capability of the cloud (10,000 GHz) mathematically compensates the propagation delay of the WAN and keeps the average latency constant (around 80-95 ms) at saturation with the highest density of devices. In addition, the strong secure task packing guarantees the fact that such data is intrinsically safe against intrusion, demonstrating the possibility of high safety, low latency, and low cost to coexist dynamically.

7.3. Energy Consumption Impact

The EASO algorithm had good secondary effects on the device battery life; although the primary goal activity was energy minimization. Local-Only execution consumed a lot of the mobile device battery as the CPU was constantly used at full speeds. Since EASO decrypts computationally intensive processes to the edge or cloud- and since radio frequency (RF) transceiver on the mobile device consumes less time in the high-power active state due to the reduction in transmission time through aggressive data compression. As a result, the overall energy usage of mobile devices was decreased by nearly a quarter of the Local-Only implementation.

7.4. Comparative Performance Summary

The comprehensive comparison across all metrics is summarized in Table 3, highlighting the multi-dimensional advantages of the EASO framework.

Table 3. Comparative Performance Summary of Different Offloading Strategies.

Metric	*Local-Only*	*Cloud-Only*	*Edge-Greedy*	*EASO*
Bandwidth Cost	*0*	*Very High*	*High*	***Low (40% reduction)***
Average Latency	*Very High*	*Moderate*	*Variable*	***Optimal (sub-100ms)***
Energy Consumption	*Very High*	*Low*	*Moderate*	***Low (22% reduction)***
Security Compliance	*High*	*Moderate*	*Moderate*	***High (Encrypted)***
Scalability	*Limited*	*High*	*Moderate*	***High***

8. Limitations and Future Work

Although the suggested EASO framework illustrates important achievements in the field of cost optimization and safe task offloading of bandwidth, the current study is carried out within the definite architectural constraints that introduce the limitation to much larger real-life applications. **On Limitations:** The paper does not investigate in depth the limitations of the proposed approach in the highly volatile or disconnected settings, but mentions the dire necessity of additional longitudinal studies in the optimization of the bandwidth expenses in mobile cloud application outsourcing. More importantly, the research approach presupposes the realistic yet controlled deployment setup with a stable, heterogeneous network-less environment with mobile devices with a baseline of processing resources (1.0-2.5 GHz CPU) and edge servers structurally situated in the low-latency and a single-hop proximity (usually less than 10-20ms RTT).

8.1. High-Mobility Scenarios

The continuous, low-latency proximity-based assumption of a single edge server cannot be assumed in highly mobile networks, e.g., high-speed vehicular ad-hoc networks (VANETs) or swarms of drones. The mobility of the high velocity would result in frequent handovers between the edge nodes and in no case would not require intermediate task data re-encryption and re-transmission that would result in zero bandwidth savings made in the stationary task packing model [22]. Strategy updates through the multiple forward rounds are needed to converge the game-theoretic to Nash Equilibrium. Where network topology can vary within milliseconds (as is the case with vehicle-to-vehicle communications on a highway), convergence time can be prohibitively lengthy to give timely offloading decisions.

 doi:10.20944/preprints202603.0270.v1

8.2. Heterogeneous Network Conditions

The existing model presupposes rather stable 5G cellular connectivity and predicts channel characteristics. The movement between heterogeneous network vendors (Wi-Fi, 4G LTE, 5G mmWave, satellite links, etc.) with all being very different bandwidth, latency, and reliability characteristic profiles is common practice on mobile devices. Dynamic network-aware adjustment of the bandwidth pricing parameters would be required by the EASO algorithm Φ_{edge} and Φ_{cloud} in real-time based on the active network interface.

8.3. Partial Task Offloading and Dependency Graphs

The formulation that is currently in place assumes that each activity can be only performed as an atomic unit, that is, it has to be carried out in a single point. Most practical systems are comprised of complex calculation systems that are modelled as Directed Acyclic Graphs (DAGs), with tasks that depend on one another and can be executed in multiple levels [13]. Implementing EASO as a tool to carry out fine-grained task partitioning and managing the inter-tasks dependencies would make EASO much more applicable to practices such as augmented reality, real-time video analytics, and distributed machine learning inference.

8.4. Future Research Directions

With these bounds in mind, next-generation systems can be much more valuable through highly dynamic, hybrid approaches, that retains the enormous processing capabilities and scalability to the world of cloud-only offloading, and, at the same time, uses edges computing to combat the natural limitations of latency in the cloud. Research directions that may be developed in the future include:

1. **Deep Reinforcement Learning Convention**: Substitute the iterative game-theoretic player with the Deep Q-Networks (DQN) or Actor-Critic systems that can obtain optimal offloading choices among historical network information, that is, with later decisions that are zero-converging [17].
2. **Federated Learning to Collaborative Optimization**: Allow multiple edge servers to interact in the joint training of common offloading models, without sharing raw user data shops and retain privacy without increasing the quality of international decisions [23].
3. **Quantum-Resistant Cryptography**: With the development of quantum computing, the existing AES-256 encryption will likely be compromised. Exploring the future of the security layer, post-quantum cryptographic algorithm in EASO will be investigated.
4. **Multi-Objective Pareto Optimization**: EASO is extended in this task to consider bandwidth cost, latency, energy, and security as a multi-objective maximization problem to generate Pareto-optimal frontiers offering administrators chances to pick the operating points depending on the dynamic policy needs.

9. Conclusion

The increased use and spread of mobile computing, along with the subsequent increase in data, have taken the traditional cloud computing infrastructures to the outer limits of bandwidth capacity. The paradigm of viewing the transmission in the network as an endless and free resource is inherently flawed in cost-sensitive and bandwidth-constrained systems, which causes serious network congestion and unsustainable spending on operations.

This research proposed a three-level hierarchical system framework that integrates mobile devices, edge servers and the centralized cloud platforms, in an intricate manner, into a single computing system. We managed to create the Edge-Aware Secure Offloading (EASO) algorithm by devising the bandwidth consumption variable as a direct financial cost variable but not as a physical constraint itself. This hybrid method is the only to dynamically place computational work with surgical precision, and adaptive data compression, the principles of secure multi-party computation, and game-theoretic optimization, are comprised.

 doi:10.20944/preprints202603.0270.v1

The outcome of the experiment explicitly confirmed the effectiveness of our method through strict simulation tools by using realistic network structures and pricing models. The EASO framework recorded an unbelievable reduction in overall bandwidth expenses by their traditional methodologies of cloud-centric offloading up to 40 percent, and also enforced rigid latency constraints of under 100 ms on real-time applications. More so, the system used a mathematically modelled security overhead (through encrypted and coded packing of tasks) to provide strong data confidentiality and integrity as well as ensuring the performance constraints.

The secondary advantages of lowering the energy consumption of mobile devices (22% less than local-only execution) and better scalability at high levels of device stacking further highlight the overall holistic value of the EASO system. This is likely to become the most critical parameter in the global network infrastructure as the digital environment continues moving into the 5G, 6G, and pervasive IoT environment, such bandwidth-conscious optimization models will play the largest role in ensuring the economic sustainability and operational effectiveness of the network infrastructure.

Author Contributions: Conceptualization, Buchanagandi E. Nyamajeje. and Lawrence Kerefu.; methodology, Buchanagandi E. Nyamajeje.; software, Buchanagandi E. Nyamajeje.; validation, Buchanagandi E. Nyamajeje., Lawrence Kerefu. and Huiqun Yu.; formal analysis, Buchanagandi E. Nyamajeje.; investigation, Lawrence Kerefu resources, Huiqun Yu.; data curation, Huiqun Yu.; writing—original draft preparation, Buchanagandi E. Nyamajeje.; writing—review and editing, Buchanagandi E. Nyamajeje.; visualization, Buchanagandi E. Nyamajeje; supervision, Huiqun Yu.; project administration, DarTU; funding acquisition, DarTU. All authors have read and agreed to the published version of the manuscript.

Funding: We are grateful to the DarTU management of University for their dedication to developing intellectual study. The university's deployment of specific funds to support research activities. Also, this work was partially supported by the NSF of China under grants No. 61173048 and No. 61300041, Specialized Research Fund for the Doctoral Program of Higher Education under grant No. 20130074110015, and the Fundamental Research Funds for the Central Universities under Grant No.WH1314038.

Institutional Review Board Statement: "Not applicable" for studies not involving humans or animals.

Informed Consent Statement: "Not applicable." for studies not involving humans.

Acknowledgments: We are grateful to the DarTU management of University for their dedication to developing intellectual study. The university's deployment of specific funds to support research activities. Also, this work was partially supported by the NSF of China under grants No. 61173048 and No. 61300041, Specialized Research Fund for the Doctoral Program of Higher Education under grant No. 20130074110015, and the Fundamental Research Funds for the Central Universities under Grant No.WH1314038

Conflicts of Interest: "The authors declare no conflicts of interest.

References

1. Abreha, H. G., Hayajneh, M., & Serhani, M. A.. Federated Learning in Edge Computing: A Systematic survey. Sensors, (2022), 22(2), 450. https://doi.org/10.3390/s22020450
2. Acheampong, A., Zhang, Y., Xu, X., & Kumah, D. A.. A review of the current task offloading algorithms, strategies and approach in edge computing systems. Computer Modeling in Engineering & Sciences, (2022), 134(1), 35–88. https://doi.org/10.32604/cmes.2022.021394
3. Akherfi, K., Gerndt, M., & Harroud, H. . Mobile cloud computing for computation offloading: Issues and challenges. Applied Computing and Informatics, (2016)., 14(1), 1–16. https://doi.org/10.1016/j.aci.2016.11.002
4. Alharbi, H. A., Aldossary, M., Almutairi, J., & Elgendy, I. A. Energy-Aware and secure task offloading for Multi-Tier Edge-Cloud computing systems. Sensors, . (2023)., 23(6), 3254. https://doi.org/10.3390/s23063254
5. Almuseelem, W. Deep Reinforcement Learning-Enabled Computation Offloading: a novel framework to energy optimization and Security-Aware in vehicular Edge-Cloud computing networks. Sensors, (2025), 25(7), 2039. https://doi.org/10.3390/s25072039

 doi:10.20944/preprints202603.0270.v1

6. Avan, A., Azim, A., & Mahmoud, Q. H. . A State-of-the-Art Review of Task Scheduling for Edge Computing: A Delay-Sensitive Application Perspective. Electronics, (2023), 12(12), 2599. https://doi.org/10.3390/electronics12122599
7. Azamuddin, W. M. H., Aman, A. H. M., Sallehuddin, H., Salam, M., & Abualsaud, K. (2024). Mathematical Models for Named Data Networking Producer Mobility Techniques: A review. Mathematics,(2024),12(5),649. https://doi.org/10.3390/math12050649
8. Carvalho, A., Riordan, D., & Walsh, J.. A Novel Edge Platform Streamlining Connectivity between Modern Edge Devices and the Cloud. Future Internet, (2024), 16(4), 111. https://doi.org/10.3390/fi16040111
9. Cheng, S., Ren, T., Zhang, H., Huang, J., & Liu, J.. A Stackelberg-Game-Based framework for edge pricing and resource allocation in mobile edge computing. IEEE Internet of Things Journal, (2024), 11(11), 20514–20530. https://doi.org/10.1109/jiot.2024.3372016
10. Edje, A. E., Latiff, A., & Chan, W. H.. IoT data analytic algorithms on edge-cloud infrastructure: A review. Digital Communications and Networks,(2023), 9(6), 1486–1515. https://doi.org/10.1016/j.dcan.2023.10.002
11. G, M. F., & L, S.. Efficient task scheduling and computational offloading optimization with federated learning and blockchain in mobile cloud computing. Results in Control and Optimization, (2025), 18, 100524. https://doi.org/10.1016/j.rico.2025.100524
12. Gilbert, C., Gilbert, M. A., & Jnr, M. D. Secure data management in cloud environments. International Journal of Research and Innovation in Applied Science, (2025), IX(IV), 25–56. https://doi.org/10.51584/ijrias.2025.10040003
13. Guo, R., Zhou, L., Li, L., Song, Y., & Xie, X..Dependent Task Graph offloading model based on deep reinforcement learning in mobile edge computing. Electronics, (2025).,14(16), 3184. https://doi.org/10.3390/electronics14163184
14. Hwaitat, A. K. A., Almaiah, M. A., Ali, A., Al-Otaibi, S., Shishakly, R., Lutfi, A., & Alrawad, M.. A New Blockchain-Based Authentication Framework for Secure IoT Networks. Electronics, (2023), 12(17), 3618. https://doi.org/10.3390/electronics12173618
15. Kanellopoulos, D., Sharma, V. K., Panagiotakopoulos, T., & Kameas, A. (2023). Networking Architectures and protocols for IoT applications in smart Cities: Recent developments and perspectives. Electronics, (2023), 12(11), 2490. https://doi.org/10.3390/electronics12112490
16. Liu, C., & Sun, Z.. A Multi-Agent reinforcement Learning-Based Task-Offloading strategy in a Blockchain-Enabled edge computing network. Mathematics, (2024), 12(14), 2264. https://doi.org/10.3390/math12142264
17. Ma, Y., Zhao, Y., Hu, Y., He, X., & Feng, S.. Multi-Agent Deep Reinforcement Learning for Joint Task Offloading and Resource Allocation in IIoT with Dynamic Priorities. Sensors, (2025), 25(19), 6160. https://doi.org/10.3390/s25196160
18. Molokomme, D. N., Onumanyi, A. J., & Abu-Mahfouz, A. M. . Edge Intelligence in Smart Grids: a survey on architectures, offloading models, cyber security measures, and challenges. Journal of Sensor and Actuator Networks, (2022), 11(3), 47. https://doi.org/10.3390/jsan11030047
19. Montoya, O. D., Grisales-Noreña, L. F., & Florez-Cediel, O. D.. Exact Mixed-Integer nonlinear programming formulation for conductor size selection in balanced distribution networks: Single and Multi-Objective analyses. Electricity, (2025), 6(1), 14. https://doi.org/10.3390/electricity6010014
20. Pradeep, S., Sharma, Y., Verma, C., Sreeram, G., & Rao, P. H. RETRACTED: Express data processing on FPGA: Network interface cards for streamlined software inspection for packet processing. Applied System Innovation, (2023). , 6(1), 9. https://doi.org/10.3390/asi6010009
21. Ravisankar, S., & Maheswar, R. SecureEdge-MedChain: a Post-Quantum blockchain and federated learning framework for Real-Time predictive diagnostics in IOMT. Sensors, (2025), 25(19), 5988. https://doi.org/10.3390/s25195988
22. Rovira-Sugranes, A., Razi, A., Afghah, F., & Chakareski, J. A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook. Ad Hoc Networks, (2022), 130, 102790. https://doi.org/10.1016/j.adhoc.2022.102790
23. Sah, D. K., Vahabi, M., & Fotouhi, H. Federated learning at the edge in Industrial Internet of Things: A review. Sustainable Computing Informatics and Systems, (2025), 46, 101087. https://doi.org/10.1016/j.suscom.2025.101087

 doi:10.20944/preprints202603.0270.v1

24. Sahinidis, N. V. Mixed-integer nonlinear programming 2018. Optimization and Engineering, (2019), 20(2), 301–306. https://doi.org/10.1007/s11081-019-09438-1
25. Thabet, S., Ateya, A. A., ElAffendi, M., & Abo-Zahhad, M.. MEC and SDN Enabling Technologies, Design Challenges, and Future Directions of Tactile Internet and Immersive Communications. Future Internet, (2025), 17(11), 494. https://doi.org/10.3390/fi17110494
26. Thai, T. V., Le, M. T., Nguyen, H. V., Shin, O., & Di Benedetto, M. Next-generation MIMO empowered mobile edge computing: A comprehensive survey toward 6G systems. Ad Hoc Networks, (2025), 182, 104095. https://doi.org/10.1016/j.adhoc.2025.104095
27. Vatsavayi, V. K., & Bommireddy, D. R. Secure and efficient block cipher mode design for parallel processing and reliable security. Cryptography, . (2026), 10(1), 13. https://doi.org/10.3390/cryptography10010013
28. Wang, Q., Guo, S., Liu, J., & Yang, Y. Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing. Sustainable Computing Informatics and Systems, (2019), 21, 154–164. https://doi.org/10.1016/j.suscom.2019.01.007
29. Zhang, Y., Tang, B., Luo, J., & Zhang, J.. Deadline-Aware dynamic task scheduling in Edge–Cloud Collaborative Computing. Electronics, (2022), 11(15), 2464. https://doi.org/10.3390/electronics11152464
30. Zhou, L., Yin, H., Zhao, H., Wei, J., Hu, D., & Leung, V. C.. A comprehensive survey of artificial intelligence applications in UAV-Enabled wireless networks. Digital Communications and Networks, (2024), https://doi.org/10.1016/j.dcan.2024.11.005
31. Zou, G., Liu, Y., Yang, S., Hu, S., Gan, Y., & Zhang, B. TEDC: Temporal-aware Edge Data Caching with Specified Latency Preference. Researchgate, (2024), 822–832. https://doi.org/10.1109/icws62655.2024.00101