Cluster Computing (2026) 29:145
https://doi.org/10.1007/s10586-026-05950-z

Secure computation offloading for data leakage prevention in digital twin edge networks

Müge Erel-Özçevik[1] · **Elif Bozkaya-Aras**[2]

Received: 24 April 2025 / Revised: 3 January 2026 / Accepted: 14 January 2026
© The Author(s) 2026

Abstract

Secure computation offloading is challenging because sensitive data can be exposed to malicious users, especially when the risk of data leakage arises during transmission or processing in untrusted edge environments. Therefore, in this paper, we present a novel content-aware data leakage prevention framework that classifies computational tasks into low, medium, and high sensitivity levels. We propose to address the trade-off between energy consumption and privacy under time constraints by integrating digital twin technology for dynamic and real-time network control. Compared with traditional edge computing and IoT environments, digital twin networks provide a virtualized, continuously synchronized representation of physical entities and enables more precise, context-aware decision-making for task offloading and security management. This bidirectional mapping between the physical and virtual worlds allows for proactive risk detection, adaptive privacy control, and optimized resource allocation, which are not achievable in conventional edge or IoT systems. In this regard, we utilize the digital twin's real-time, synchronized representation of physical entities to design an energy and privacy-aware task scheduling scheme and minimize data leakage risks in digital twin edge networks. To achieve these, we analytically model both energy consumption and task processing delay using the M/M/1 queuing model. Furthermore, we propose an energy and privacy-aware genetic algorithm-based task scheduling algorithm designed to minimize energy consumption and latency while maximizing privacy in digital twin edge networks. Under strict latency constraints, numerical results show that while the conventional approach can handle up to 400 tasks, the proposed model scales efficiently to 600 tasks by prioritizing energy efficiency and privacy, and improves scalability by 1.5 times. Here, increasing the number of iterations has a much more positive effect than increasing the number of chromosomes. It achieves at most 84% privacy gain and acceptable 0.197 watts energy consumption in the topology with 0.58 second response time. In addition, the proposed approach offers the ability to further improve processing time and privacy gains through digital twin monitoring.

Keywords Digital twin · Internet of things · Edge computing · Task scheduling · Data privacy · Data leakage prevention

1 Introduction

With the proliferation of Internet of Things (IoT) devices and the exponential growth of data generated by IoT devices, ensuring timely and secure computation has become increasingly critical and the need for more intelligent data processing models has become evident [1]. Edge cloud environments have emerged as a promising paradigm that enables the distribution of computation across user devices, nearby edge nodes, and centralized cloud servers. This approach not only reduces latency and relieves network congestion, but also improves energy efficiency by balancing data processing between remote servers and local devices, as continuous data transmission to remote servers

✉ Müge Erel-Özçevik
muge.ozcevik@cbu.edu.tr

Elif Bozkaya-Aras
elif.bozkaya@msu.edu.tr

[1] Department of Software Engineering, Manisa Celal Bayar University, Manisa, Turkey

[2] Department of Computer Engineering, National Defence University Turkish Naval Academy, Istanbul, Turkey

Published online: 12 February 2026

can cause high energy consumption [2]. However, such an edge-cloud architecture introduces significant challenges related to data privacy and data leakage prevention. These challenges become critical concerns, especially in scenarios where sensitive user information (such as health records, biometric data, or location history) is processed in remote data centers.

In the application layer of IoT devices, the one of the significant data security requirement is the sensitive information protection near the providing robustness, self healing, reliability, real time, data freshness/composition, heterogeneity, interoperability/light-weight solutions/end-to-end security/privacy [3, 4]. The most common attack on this layer is privacy leakage. In the traditional edge cloud-centric model, sensitive tasks are often exposed to the risk of data leakage during transmission and processing [5, 6]. According to the European Union General Data Protection Regulation (GDPR), there are different types of data: public (low sensitive data that can be used by anyone; for example sharing traffic sensor data and weather forecast data for smart traffic management), internal (medium sensitive data that is used internally; for example personnel location data for smart office management system), confidential and restricted (highest sensitive data that can be used only by limited access; for example personal health and identity for smart health management) [7]. These are protected by a different security certification for IoT standards: Level 0 certification includes low-sensitive public data, Level 1 certification provides the minimum security, privacy and supply chain requirements for internal data, Levels 2 and 3 provide extra security requirements relating to hardware and software for the protection of intellectual property for internal and confidential data [8].

Thus, in this paper, we present a content-aware data leakage prevention framework that dynamically determines the optimal task scheduling strategy to define where each task will be processed depending on its sensitivity level. As seen in Fig. 1, we categorize computation tasks into three sensitivity level, low, medium, and high, based on the potential impact of data leakage. High-sensitivity tasks can be processed on the local device to minimize the risk of data leakage while medium and low-sensitivity tasks can be offloaded to remote servers. This categorization enables a more precise and effective leakage prevention policy and incorporates sensitivity-aware processing decisions by preventing unauthorized access and allowing sensitive data to be processed closer to its source. However, executing all high sensitive computation tasks locally is not feasible due to resource constraints and the need to meet strict latency and energy efficiency [9]. At the same time, offloading sensitive tasks to remote servers can significantly increase the risk of data leakage. Thus, to balance data privacy with computational efficiency, we propose a sensitivity-based task scheduling algorithm for edge-cloud environments with the assistance of digital twin technology.

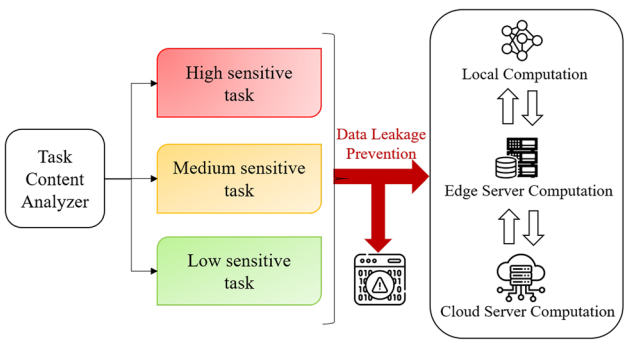

Fig. 1 An illustration of task sensitivity categorization for data leakage prevention. We categorize computation tasks according to their sensitivity level to prevent data leakage by allowing sensitive data to be processed closer to its source

The integration of digital twin with edge-cloud environments offers a dynamic and intelligent layer for real-time monitoring, decision-making, and optimization of physical systems [10]. The digital twin serves as a real-time and virtual representation of the physical system and is capable of continuously reflecting the state of the physical entities, including local devices, edge and cloud servers [11]. In our framework, the digital twin is used to track data sensitivity levels, resource availability, and latency constraints throughout the system. This supports content-aware task scheduling, where decisions are not only based on static sensitivity classifications, but also on real-time factors such as energy consumption or network constraints. By integrating digital twin technology into our model, we achieve a privacy-aware and genetic algorithm-based task scheduling algorithm that can prevent users from being eavesdropped on and minimize the risk of data leakage, especially for highly sensitive tasks.

As a result, in this paper, we propose a content-aware data leakage prevention framework in digital twin edge networks. Our proposed model not only improves data privacy by minimizing the risk of data leakage, but also optimizes energy-efficient task scheduling for the end user-edge-cloud hierarchy under the latency constraint. With the integration of the digital twin into our model, the system gains real-time awareness of device status, network conditions, and task sensitivity by allowing it to make content-aware scheduling decisions. This approach significantly reduces the risk of data leakage while maintaining low energy consumption and task processing latency [12].

The main contributions of this paper are summarized as follows:

- We present a three-layer architecture for a novel content-aware data leakage prevention framework in the digital twin edge networks.
- We classify computation tasks into low, medium, and high sensitivity levels, enabling content-aware decisions for processing at the device, edge, or cloud level. We also formulate both energy consumption and task processing time using the M/M/1 Markov model.
- We introduce an energy and privacy-aware genetic algorithm-based task scheduling algorithm to minimize the energy consumption and task processing time, and maximize data privacy. By integrating real-time insights from the digital twin, our model ensures that each task is offloaded to the user device or remote servers according to its privacy requirements and system conditions, achieving a privacy-preserving solution.
- Extensive simulations considering different methods such as time based, energy based, privacy based, privacy/energy based and proposed energy and privacy-aware genetic algorithms under time constraint are conducted to show that our proposed model effectively enhThe ances data privacy while reducing the energy consumption and task processing time by maintaining the feasibility of the solution under increased task numbers.

This paper is organized as follows. In Section 2, we review the related works. In Section 3, we introduce the content-aware data leakage prevention architecture for digital twin edge networks. In Section 4, we formalize the system model and define the energy-efficient task scheduling problem with the proposed privacy-aware and genetic algorithm-based task scheduling algorithm. We then evaluate its performance through simulations in Section 5. Finally, we conclude the paper in Section 7.

2 Related work

This section provides an overview of existing studies related to the edge-cloud security and task offloading in digital twin edge networks, focusing on energy-efficient and secure task offloading models.

2.1 Secure task offloading

MEC network presents significant security challenges, particularly the risk of data leakage to eavesdroppers. Secure task offloading is of great importance in protecting users' private information and ensuring efficient resource utilization. The existing studies have proposed various approaches to address the trade-off between security and performance in MEC, including the use of physical-layer security [13], blockchain [14], encryption techniques [15], cooperative jamming [16], and federated learning models [17, 18]. These studies aim to develop solutions that minimize delays, optimize energy consumption, and enhance security in a resource-constrained environment.

In [19], the authors explore using TrustZone-enabled edge clouds to enhance security in edge computing and to provide hardware isolation for offloaded tasks. ARM's TrustZone is a hardware-based security solution to create a secure execution environment. In the proposed framework, tasks are offloaded to edge servers equipped with ARM processors. Then, the authors propose a priority-aware secure task offloading algorithm to minimize total task completion time within an energy budget. In [20], the authors address security challenges in fog computing, where IoT devices offload tasks to nearby fog nodes to meet latency and computation needs. Traditional node selection policies assume that all participating fog nodes are trusted. However, this is not valid, especially in a cooperative environments. To solve this, the authors propose a secure task-offloading framework using blockchain-based smart contracts. This approach allows fog nodes to offload tasks securely to neighboring nodes even without prior trust. The framework ensures that untrusted or unauthorized nodes cannot accept or perform offloaded tasks. In [21], the authors investigate secure offloading in a wireless-powered MEC system and propose a physical layer security-assisted scheme, where a power beacon acts as a cooperative jammer to disrupt eavesdroppers. The goal is to maximize secrecy energy efficiency (SEE) by optimizing factors such as transmit power, time allocation, and task partitioning while ensuring secrecy and energy constraints. They develop a two-layer algorithm using variable substitutions, successive convex approximation, and the Dinkelbach method to solve the problem. In [22], the authors address the challenge of distributed denial of service (DDoS) attacks on mobile cloudlet networks, where malicious users can overwhelm cloudlet servers and disrupt services for legitimate users. To overcome this challenge, the authors propose a collaborative task offloading scheme based on balls-and-bins theory. They aim to efficiently balance task loads with minimal data, reducing the risk of overloaded cloudlets and mitigating DDoS attacks. In [16], the authors investigate the problem of secure computation offloading in MEC networks, which face limited server storage and security risks due to open wireless channels. They consider a scenario with an eavesdropper and propose a cooperative jamming-assisted scheme, where a friendly jammer disrupts potential eavesdropping. The authors formulate a delay minimization problem that jointly optimizes the service caching, transmit power, and offloading decisions and solve it using a layered algorithm combining successive convex approximation and a cross-entropy

technique. In [23], the authors discuss the security risks of wireless offloading in MEC, where data is vulnerable to eavesdropping. While physical layer security can be applied to defend against eavesdropping, this approach may directly increase energy and delay costs for mobile devices. To address this, the authors propose a secure offloading scheme in which the edge server uses full-duplex communication to broadcast jamming signals and suppress self-interference. In our work, we consider an active threat model where malicious nodes may attempt to inject data at the network edge during task offloading. Our primary focus is on securing wireless links by maximizing data privacy throughout the system.

Despite the promising capabilities of edge computing–based digital twin networks, the distributed nature of data processing and transmission introduces significant security vulnerabilities, where any data leakage can result in severe ethical, privacy, and trust issues. In this regard, in [17], the authors discuss the security and privacy challenges in digital twin systems. While AI-generated content can enhance synchronization and realism between physical and virtual twins, it introduces risks such as data leakage, model poisoning, unauthorized access, and inaccurate or manipulated digital representations. In order to overcome these challenges, the proposed framework combines federated learning and semantic communication and aims to mitigate these threats by improving data confidentiality, model integrity, and communication efficiency. Similarly, in [24], the authors explore an immersive communication framework for digital twin systems, which integrates edge computing and tactile Internet technologies to achieve real-time, reliable, and interactive connections between the physical and virtual worlds. The framework aims to deliver ultra-low latency, high-fidelity feedback, and seamless multi-modal data exchange. In [25], the authors investigate the potential of AI-driven digital twins to advance IoT-based healthcare by enabling real-time, high-fidelity digital replicas of individuals for personalized monitoring, diagnosis, and treatment. However, since the system may consist of highly sensitive individual-level data that reflects a person's physical and health status, any data leakage could lead to severe ethical and moral concerns. Therefore, developing robust security and privacy protection mechanisms is crucial in digital twin edge systems.

Although several studies have addressed data privacy protection in MEC, they do not focus on the varying levels of privacy requirements associated with different types of computation tasks. Specifically, the proposed solutions typically fail to account for the distinct needs of high-sensitive, medium-sensitive, and low-sensitive tasks. This paper aims to fill this gap by focusing on privacy challenges within the edge-cloud architecture with the assistance of digital twin implementation. We propose an energy-efficient task offloading strategy that not only enhances data privacy but also optimizes energy cost, effectively addressing privacy concerns across tasks with varying sensitivity levels.

2.2 Edge-cloud security in digital twin systems

Considering the integration of the physical and digital worlds, data leakage may lead to the disclosure of sensitive information and damage to data privacy [26, 27]. Therefore, protecting these data is crucial to maintain trust and data security in the digital twin system.

In [28], the authors aim to optimize resource monitoring and management in AI systems through a secure and efficient digital twin platform. They introduce a distributed digital twin cloud that integrates blockchain technology to enhance data security via cryptographic algorithms. The platform is evaluated for task scheduling and data consistency. In [29], the authors introduce an intelligent secure task offloading and caching scheme for digital twin-assisted vehicular edge computing. They formulate a joint optimization problem involving blockchain, communication, computation, and caching to minimize latency and improve transaction throughput. Due to the problem's complexity, it is modeled as a Markov Decision Process and solved using a multi-agent double actor-critic algorithm. In [30], the authors present an AI-enhanced digital twin framework to improve security and computational efficiency in 6G Internet of Vehicles networks and also aim to reduce system latency and energy consumption while improving the attack detection rate. They address the challenge of centralized computation since this can lead to unstable attack detection. To overcome this challenge, the proposed framework introduces a feature engineering module to reduce data dimensions and distribute computational load across layers. It also integrates an online learning-based and network-aware attack detection system.

Recent researches also emphasize the importance of edge–cloud collaboration in enabling efficient deployment and optimization of digital twins, where detailed modeling analyses address critical challenges such as dynamic task offloading, resource allocation, and adaptive model integration between the edge and cloud environments [31, 32]. In [33], the authors propose an online optimization framework to deploy digital twins within an end–edge–cloud collaborative architecture and aim to balance performance, energy, and delay constraints. They focus on modeling and optimizing task offloading, communication, and computation allocation under dynamic and uncertain conditions such as user mobility. To efficiently manage these challenges, the authors introduce the two-timescale accuracy-aware online optimization algorithm by coordinating decisions between the edge and cloud. In [18], the authors introduce a federated

edge–cloud collaborative framework to build self-evolving digital twin models. The proposed system decomposes a global digital twin at the cloud into multiple partial digital twins at edge servers, which are continuously updated using locally collected sensor data and then integrated back into the global model. To optimize digital twin deployment and task offloading, the authors formulate an online optimization problem that jointly manages digital twin assignments, sensor associations, and resource allocation. In [34], the authors present an edge-based digital twin architecture to develop digital replicas of portable and wearable sensors and process historical and real-time data efficiently. This edge-assisted digital twin design reduces latency, alleviates data transfer constraints, and enhances decision-making and classification accuracy.

Although these studies have explored various methods to enhance edge-cloud collaboration and security in digital twin systems, centralized processing models expose data to more potential attacks during transmission and storage. To address this concern, in this paper, we address the need to increase data security by reducing reliance on centralized systems. We propose that moving data processing closer to the source through edge computing can minimize the risk of exposure to potential leaks by reducing the amount of sensitive data transmitted to centralized systems.

3 System architecture

In this section, a description of the system architecture for the digital twin edge network is presented. The digital twin edge network is comprised of three layers: the physical layer, the digital twin layer, and the service layer. The physical layer encompasses all network entities, including IoT devices, MEC servers, and cloud servers. The IoT devices are configured to connect to the nearest base station that is equipped with a MEC server[35]. MEC servers are under the control of one cloud server. Each IoT device generates computation tasks. The data privacy sensitivity of these tasks $Task_{i,j,k}$ is defined as $p_{i,j,k}$, where i, j, and k show the indices of local devices, MEC servers, and cloud server, respectively. In this study, we adopt an active threat model in which malicious nodes may try to tamper with or inject malicious data at the network edge during the process of task offloading. Our main concern lies in securing the communication links and ensuring that data privacy is maximized by preventing any unauthorized access or manipulation in the system. Therefore, we aim to protect sensitive information from potential breaches and maintain the security of the digital twin environment.

In Fig. 2, the interconnection of all layers is illustrated in the context of offloading a task from an IoT device to a cloud server. In the digital twin layer, digital twins are created for each network entity. Here, they are the virtual representation of the physical devices [36]. The digital twin layer continuously aggregates data from IoT devices, including the size of computation tasks and data privacy sensitivity. Then, it calculates energy and time metrics, which are formulated with our proposed Markov model. As a result, it creates a virtual network to follow the real network status, which helps to improve system performance [37]. Here, different task scheduling approaches are virtually applied in the digital twin layer, which are detailed in the service layer in subsection 4.2.

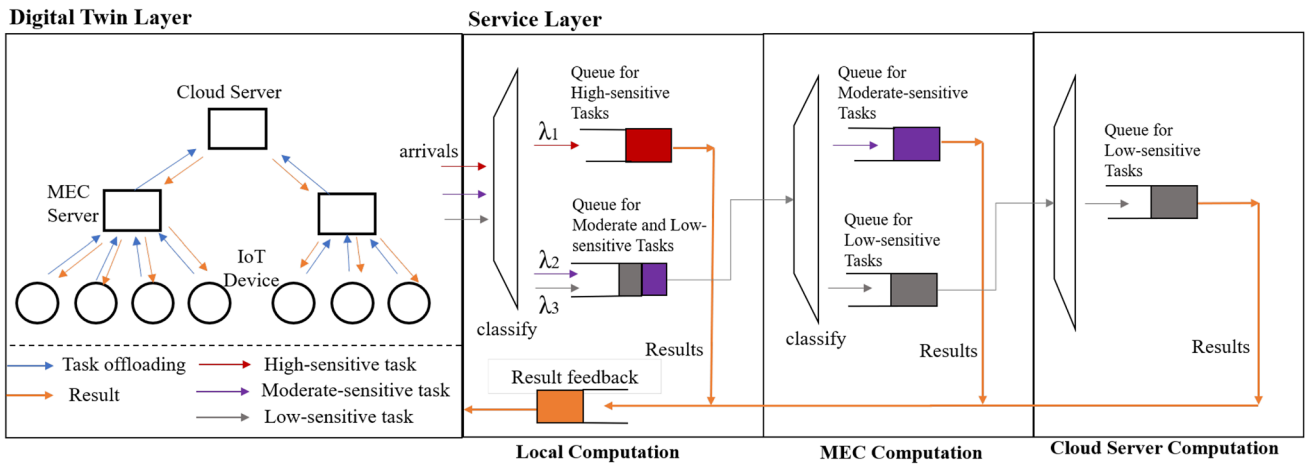

Fig. 2 The implementation of DT-assisted data privacy prioritized task scheduling with user device-edge-cloud collaboration and illustration of the interconnection of layers for offloading high, moderate, and low-sensitive tasks from IoT devices to MEC and cloud servers. The arrows indicate the data flow between the layers based on energy and privacy requirements. In the service layer, the tasks are classified, and various task scheduling approaches are applied. Then, result feedback is sent to the digital twin layer to optimize task offloading decisions and improve overall system performance. The digital twin layer calculates energy and time metrics using the proposed model and generates a virtual network to mirror real-world conditions

In the service layer, the proposed task scheduling algorithm is executed. This algorithm takes into account the privacy sensitivity of tasks and the energy metric under time constraints. As seen in Fig. 2, the computational tasks are divided into three categories, namely high-sensitive, medium-sensitive, and low-sensitive data privacy classifications. They are modeled by using an M/M/1 queuing system. It has been determined that the use of this model facilitates the theoretical establishment of a multi-objective model for task scheduling. The objective of the task scheduling model is twofold: First, it aims to minimize energy and second, it aims to maximize data privacy under time constraint. In the context of local computation, tasks exhibiting high sensitivity are accorded priority, while those of medium and low sensitivity can be assigned to the MEC. In the context of MEC computation, tasks exhibiting a medium level of sensitivity are accorded a higher priority, while those of a lower sensitivity can be assigned to the cloud server. In conclusion, low-sensitive tasks can be executed in the context of cloud computing.

In Table 1, example solutions for the energy and privacy-aware task scheduling problem under time constraint are given. There are N number of tasks in the topology where energy consumption is calculated as given in Eq.5. Energy consumption for the user device varies according to the offloading decision of the task through MEC or cloud server. If the task is executed in local, the energy consumption is calculated as the product of task execution time (T_L) and the consumed power of local device (P_L) while executing task. Here, P_L is taken as constant for each task and T_L, T_M, T_C will be calculated by using M/M/1 markov model in the following subsection 4. In the event that a task is assigned to a MEC or cloud server, the energy expenditure (EE) is determined by calculating the sum of the energy consumption associated with the transmission and reception of data between the local device and the server in proximity to the consumed energy on the server during the execution of the task. Here, P_L^{tr} and P_L^{rec} are consumed power while transmitting and receiving data between local device and MEC or cloud server. D is defined as the total data amount of tasks to offload from local devices to MEC or cloud servers. R is the data transmission rate between the local device and MEC or cloud servers. P_{idle} is the consumed energy of the local device while the task is executed on either on MEC or on cloud servers. On the other hand, if the task is offloaded to MEC or cloud servers, the consumed energy while executing the task is also added to the energy consumption in the local device by the product of $T_M \cdot P_M$ or $T_C \cdot P_C$ where P_M and P_C are the consumed power of MEC and cloud servers while executing task of local IoT device, respectively.

Moreover, data privacy sensitivity is formalized as in the following equation:

$$p_{i,j,k} = \begin{cases} High, & p_{i,j,k} \geq p_L \\ Medium, & p_L > p_{i,j,k} \geq p_M \\ Low, & p_M > p_{i,j,k} \geq p_C \end{cases} \quad (1)$$

$p_{i,j,k}$ defines the data privacy sensitivity of $Task_{i,j,k}$. Here, the thresholds of data privacy sensitivity are previously listed as $p_L > p_M > p_C$. Here, p_L, requests higher data privacy than p_M. Moreover, p_M defines higher data privacy than p_C. They define high, medium, and low sensitive data privacy of tasks, respectively.

The objective of the data privacy prioritized task scheduling problem is to identify a solution that optimizes two conflicting objectives: minimizing energy consumption while executing tasks, denoted by (E), and maximizing data privacy, denoted by (P), under the task processing time (T) constraint. This is illustrated in the last row of Table 1. In the event that a task of high data privacy sensitivity is executed on a cloud (C) or a MEC (M) server, this solution is suboptimal. It is recommended that the task be executed on a local (L) device. Conversely, the execution of all tasks is not feasible on local devices, thereby increasing the processing time on a device. Consequently, tasks are scheduled between local, MEC servers, and cloud server in order to minimize T by considering data privacy sensitivities. The problem formulation and the proposed algorithm are detailed in the subsequent section.

4 Problem formulation and proposed task scheduling algorithm

In this section, we initiate the formulation of the optimization problem for privacy- and energy-aware task scheduling. The metrics are extracted by utilizing the M/M/1 Markov model. Following the delineation of the multi-objective optimization formula, a range of approaches that

Table 1 Example solutions for data privacy prioritizing task scheduling problem

Task id	Privacy Sensitivity				$\sum T$	$\sum E$	$\sum P$
	1, Low	2, Medium	...	N, High			
Case 1	L	L	.....	L	T_1	E_1	P_1
Case 2	L	M	.....	C	T_2	E_2	P_2
Case 3	C	L	.....	C	T_3	E_3	P_3
...	...	...	...	...	....	...	...
Case N (Optimal)	C	M	...	L	T_N<1 sec	E_{min}	P_{max}

seek to address this issue are examined. In conclusion, the proposed energy- and privacy-aware task scheduling algorithm is delineated in the form of pseudo code.

4.1 Problem Formulation

The multi-objective optimization of the energy and privacy-aware task scheduling problem is analytically defined as follows:

$$\min \sum_{i\in L, j\in M, k\in C} E_{i,j,k}, \quad \max \sum_{i\in L, j\in M, k\in C} p_{i,j,k} \tag{2a}$$

subject to:

$$\lambda \cdot (p_L + p_M + p_C) = \lambda \tag{2b}$$

$$p_L = \frac{\sum_{i\in L} \lambda_i}{\lambda} \tag{2c}$$

$$p_M = \frac{\sum_{j\in M} \lambda_j}{\lambda} \tag{2d}$$

$$p_C = \frac{\sum_{k\in C} \lambda_k}{\lambda} \tag{2e}$$

$$\forall i \in L, \forall i \notin M, \forall i \notin C \tag{2f}$$

$$\forall j \in M, \forall j \notin L, \forall j \notin C \tag{2g}$$

$$\forall k \in C, \forall k \notin L, \forall k \notin M \tag{2h}$$

$$\sum T_{i,j,k} \leq 1sec, \quad \forall i \in L, \forall j \in M, \forall k \in C \tag{2i}$$

where $\sum E_{i,j,k}$ is the total energy consumption in the system and it is calculated as given in Eq. 5. $p_{i,j,k}$ is a newly proposed data privacy metric which is calculated as in Eq. 7. In this context, $\sum T_{i,j,k} = T_L + T_M + T_C$ is the total system delay as the time required to process tasks according to the scheduling output of i, j, k.

In this study, the M/M/1 Markov model is employed to model local, MEC, and cloud servers. The total time for each M/M/1 system is calculated using Little's theorem as $T = \frac{N}{\lambda}$ [38] where $N = \frac{\rho}{1-\rho}$ and $\rho = \lambda/\mu$. Then, T is finalized as follows:

$$T_{L,M,C} = \begin{cases} \frac{\lambda \cdot p_L}{\mu_L - \lambda_i}, & Task_i \in L \\ \frac{\lambda \cdot p_M}{\mu_M - \lambda_j}, & Task_j \in M \\ \frac{\lambda \cdot p_C}{\mu_C - \lambda_k}, & Task_k \in C \end{cases} \tag{3}$$

where μ is the serving time and it is defined as $\mu_L < \mu_M < \mu_C$ in the edge computing system. To calculate the serving time of each server, we consider the required CPU cycle per bit for servers ($c_{L,M,C}$), the size of the task (S), and the computing capability of the server ($f_{L,M,C}$) [39]. Accordingly, the serving time formula is given as follows:

$$\mu_{L,M,C} = \frac{c_{L,M,C} \cdot S}{f_{L,M,C}} \tag{4}$$

The task scheduling solution should prioritize privacy, which is calculated as outlined in Eq. 7. In this equation, any task necessitating the highest data privacy level, denoted by p_L, can be scheduled on Cloud p'_C, which is not optimal in terms of privacy sensitivity. Consequently, the objective function for data privacy is calculated based on the true positives of local, MEC, and cloud servers, divided by the total number of tasks.

$$\sum_{i\in L, j\in M, k\in C} E_{i,j,k} = \begin{cases} T_L \cdot P_L, & Task_i \in L \\ P_L^{tr} \frac{D_{L\to M}}{R_{L\to M}} + T_M \cdot P_{idle} + P_L^{rec} \frac{D_{M\to L}}{R_{M\to L}} + T_M \cdot P_M, & Task_j \in M \\ P_L^{tr} \frac{D_{L\to C}}{R_{L\to C}} + T_C \cdot P_{idle} + P_L^{rec} \frac{D_{C\to L}}{R_{C\to L}} + T_C \cdot P_C, & Task_k \in C \end{cases} \tag{5}$$

$$\sum_{i\in L, j\in M, k\in C} T_{i,j,k} = \frac{\sum_{i\in L} \lambda_i}{\mu_L - \lambda_i} + \frac{\sum_{j\in M} \lambda_j}{\mu_M - \lambda_j} + \frac{\sum_{k\in C} \lambda_k}{\mu_C - \lambda_k} \tag{6}$$

$$\sum_{i\in L, j\in M, k\in C} p_{i,j,k} = \frac{\sum_{i\in L}(p'_i = p_L) + \sum_{j\in M}(p'_j = p_M) + \sum_{k\in C}(p'_k = p_C)}{N} \tag{7}$$

4.2 The proposed energy and privacy-aware task scheduling algorithm

In this section, the proposed genetic algorithm-based task scheduling is detailed by giving the implementation steps. We propose different scheduling approaches that are executed virtually in the service layer. These approaches are listed below:

- *The conventional task scheduling approach:* The tasks are randomly distributed in a manner that does not take into account energy and privacy sensitivity requests.
- *The conventional time-based task scheduling approach (CT):* A solution pool is to be generated at random, and the best result is to be selected in accordance with the time (T) metric, which is to be minimized.
- *The proposed time-based task scheduling approach (PT):* A solution pool is to be generated at random, and a genetic algorithm is executed with determined iteration and chromosome numbers, and the best result is to be selected in accordance with the time (T) metric, which is to be minimized.

- *The conventional privacy-based task scheduling approach (CP):* A solution pool is to be generated at random, and the best result is to be selected in accordance with the privacy (P) metric, which is to be maximized.
- *The proposed privacy-based task scheduling approach (PP):* A solution pool is to be generated at random, and a genetic algorithm is executed with the determined iteration and chromosome numbers, and the best result is to be selected in accordance with the privacy (P) metric, which is to be maximized.
- *The conventional energy-based task scheduling approach (CE):* A solution pool is to be generated at random, and the best result is to be selected in accordance with the energy (E) metric, which is to be minimized.
- *The proposed energy-based task scheduling approach (PE):* A solution pool is to be generated at random, and a genetic algorithm is executed with determined iteration and chromosome numbers and the best result is to be selected in accordance with the energy (E) metric, which is to be minimized.
- *The conventional energy and privacy-aware task scheduling approach (CP/E):* A solution pool is to be generated at random, and the best result is to be selected in accordance with the privacy/energy (P/E) metric, which is to be maximized.
- *The proposed energy and privacy-aware task scheduling approach (PP/E):* A solution pool is to be generated at random, and genetic algorithm is executed with the determined iteration and chromosome numbers and the best result is to be selected in accordance with the privacy/energy (P/E) metric, which is to be maximized.
- *The proposed energy and privacy-aware genetic algorithm-based task scheduling approach:* It is initially executed as PP/E approach until the time (T) constraint is under 1 second. To be under the feasible region, the best approach is selected between the others, and it is executed by considering the time (T) constraint as given in Alg. 1. Subsequently, the proposed task scheduling provides the optimal offloading decision in the digital twin layer, and the result is transmitted to the physical network.

Algorithm 1 Energy and Privacy-Aware Task Scheduling Algorithm

Require: $p_{i,j,k}$ and $\lambda_{i,j,k}$
Ensure: Optimal Task Scheduling ($S_{optimal}$)
function ConventionalTaskScheduling(())
Randomly distribute tasks to L, M, C ▷ O(Vx|S_i|)
Randomly change some of $S_i \in S$ ▷ O(1)
Calculate Time(S), $\sum_{i\in L,j\in M,k\in C} T_{i,j,k}$ ▷ Eq.6, O(Vx|S_i|)
Calculate Energy(S), $\sum_{i\in L,j\in M,k\in C} E_{i,j,k}$ ▷ Eq.5, O(Vx|S_i|)
Calculate Privacy(S), $\sum_{i\in L,j\in M,k\in C} p_{i,j,k}$ ▷ Eq.7, O(Vx|S_i|)
S_{best}← FindBest(S_{pool}) ▷ O(1)
Return S_{best}
end function
function Main(())
Initialize S_{pool} ▷ O(Vx|S_i|)
S_i←CONVENTIONALTASKSCHEDULING() ▷ O(Vx|S_i|)
Calculate $\sum p_{i,j,k} / \sum E_{i,j,k}$ ▷ Eqs.7, 5, O(1)
for int i=0;i<I;i++ **do** ▷ O(Ix|S_i|)
Parent Selection(S_{pool}) ▷ O(1)
Recombination(S_{pool}) ▷ O(|S_i|)
Mutation(S_{pool}) ▷ O(1)
Check constraints ▷ Eq.2a,O(|S_i|)
Calculate Fitness function ▷ Eqs.5, 6, 7,O(|S_i|)
Natural Selection(S_{pool}) ▷ O(1)
end for
Return $S_{optimal}$← FindBest(S_{pool}) ▷ O(1)
end function

In Algorithm 1, the pseudocode of the proposed energy and privacy-aware task scheduling algorithm is given. It has two main subfunctions, named conventional task scheduling and main. The main function randomly initializes the solution pool where the chromosome number is V, i.e. size of S_{pool} ($|S_{pool}|$). It calls a conventional task scheduling algorithm to calculate time, energy, and privacy metrics per chromosome and determine the best solution in a conventional way. According to the conventional task scheduling approach (CT, CP, CE, CP/E), the best solution is determined and returned to the main function. Then, the proposed genetic algorithm is executed by predefining the iteration (I), chromosome numbers (V), and total task number (size of each chromosome, $|S_i|$). It runs the following steps: Parent Selection, Recombination, Mutation, Checking constraints to be in the feasible range, and Natural Selection. Parent selection involves randomly selecting two chromosomes from the parent pool, which are the ten best solutions according to the determined metric (CT, CP, CE, CP/E). In recombination, two points are randomly selected, and the parent chromosomes are divided into three pieces. The first and last parts of the child chromosomes are taken from the first parent's pieces, and the middle part is taken from the middle piece of the second parent. In the mutation step, a point is randomly selected and mutated using a randomly generated number (0, 1, or 2), which represents the executing task in local, MEC or cloud locations in digital form. Natural selection selects the best V chromosomes, which is equal to the size of the pool ($|S_{pool}|$). This is achieved by storing the chromosomes in a priority queue as a data structure of a solution pool, which includes both newly generated candidate solutions and old parent solutions. Thanks to using this data structure, it is already sorted list in terms of the decision metric and therefore, there is no need to run sorting algorithm in each step which minimizes the complexity of the algorithm. In these steps, the algorithm considers the fitness function, which is determined by the proposed approaches (PT, PP, PE, PP/E) according to the time constraint of the digital twin layer. However, the processing time of this pseudo-code is significant for the response time of the digital twin layer, which should be under 1 minute. The total complexity of the algorithm is $O((V+I)x|S_i|)$ which directly depends on the task size and chromosome number in the solution pool. Therefore, the convergence and time performances of the proposed algorithm are evaluated with different iterations and chromosome numbers by comparing different approaches in the following Section 5.

5 Performance evaluation

In this section, an evaluation of the proposed energy and privacy-aware task scheduling approach is conducted through a comparison of different approaches. The network topology under consideration comprises *M* IoT devices, one MEC, and one cloud server. The execution of the designated task *N* is distinguished by a numerical sequence ranging from 100 to 1000. Each task size ($|S_i|$) is measured in megabytes. The chromosome number (V) ranges from 1 to 10000, and the iteration number (I) varies from 2 to 1000. The serving capability of local, MEC, and cloud servers (f_L, f_M, f_C) is assumed to be 20 seconds, 10 seconds, and 1 second, respectively. The execution of each task requires 100 cycles for each server, c_L, c_M, and c_C, as defined in Eq. 4. The performance of the proposed algorithm is compared with conventional approaches by implementing digital twin and service layers using by Java programming language. The operation is executed on a digital twin controller that operates on a Mac OS Sequoia version 15.3.2 operating system, which contains an Apple M4 processor and 16 GB of RAM.

In Fig. 3, the left y axis shows the privacy gain (%) results (bar chart) which is calculated by 100 privacy [0-1]. The right y axis (line plot) shows the time results of 9 different approaches when the task number (N=$\sum$ L+M+C)

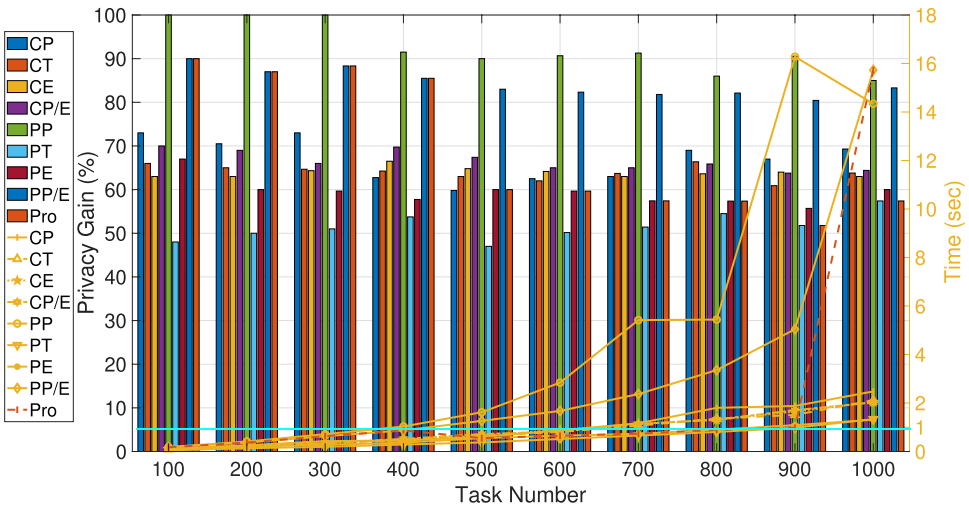

Fig. 3 Privacy Gain (%) and Time (sec) results of different approaches for different task numbers

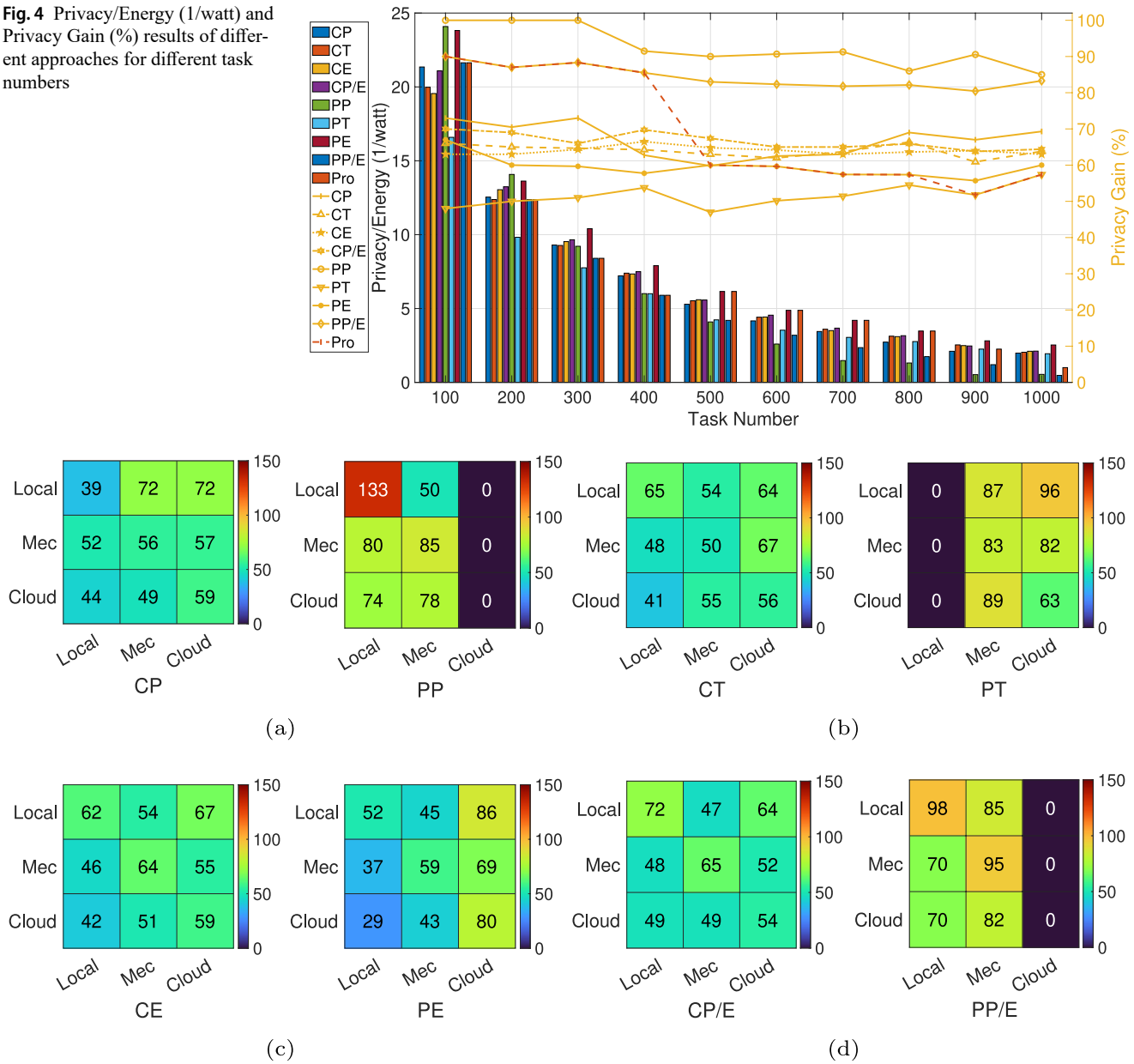

Fig. 4 Privacy/Energy (1/watt) and Privacy Gain (%) results of different approaches for different task numbers

Fig. 5 Task scheduling results for different approaches when the task number is 500 (**a**) Conventional Privacy (CP) based / Proposed Privacy (PP) based (**b**) Conventional Time (CT) based / Proposed Time (PT) based (**c**) Conventional Energy (CE) based / Proposed Energy (PE) based (**d**) Conventional Privacy/Energy (CP/E) based / Proposed Privacy/Energy based (PP/E)

increases. The time- and energy-based approaches function by allocating tasks to cloud devices with the objective of minimizing processing time, whereas privacy-based approaches schedule tasks to local devices to maximize data privacy. As the number of tasks increases, the processing time of the proposed approaches also increases when the chromosome number (V) is 100, and the iteration number (I) is 10000. In the context of a time-constrained scenario, specifically a one-minute period within the digital twin layer, the optimal approach is identified as PP/E. Consequently, the proposed algorithm employs a fitness function that prioritizes energy efficiency and privacy (PP/E) until the task number reaches 600. Subsequently, the proposed energy-based approach (PE) converges to optimal results until the task number reaches 900. Subsequently, it is determined that all approaches are impractical in this region. Therefore, the proposed algorithm functions similarly to the proposed time-based approach (PT).

In Fig. 4, the left y axis shows the privacy/energy metric (bar chart), and the right y axis (line plot) shows privacy gain (%) results of 9 different approaches when the task number increases. As the task number increases, the privacy/energy

metric undergoes a gradual decrease for all approaches. As the task number is yet to reach 500, the proposed algorithm has yielded an approximate privacy gain of 85%, which is among the top two results. When the task number is between 500 and 900, the time constraint cannot be sustained by the P/E approach. The other approaches exist, and they can ensure a higher level of data privacy. However, the proposed approach converges to a 60% level of privacy gain. It has been demonstrated that the alternative methodologies are incapable of meeting the time requirements of the digital twin layer because of the time complexity.

In Fig. 5, the ensuing dataset presents the distribution results of different task scheduling approaches when 500 tasks are present in IoT devices in total. Furthermore, the impact of the genetic algorithm on the conventional initialization of task scheduling approaches can be clearly seen when the analysis is conducted horizontally. For example, privacy-based genetic algorithm (PP) distributes all tasks into local and MEC, then, it takes privacy gain from 59% to 90% with 1.6 seconds. However, this result makes it unfeasible for the digital twin layer. Conversely, CT reduces the time from 0.65 seconds to 0.37 seconds, thereby achieving a 20% decrease in data privacy gain by allocating tasks to remote servers exclusively, as opposed to IoT devices. According to the energy-based approach, the execution of the genetic algorithm results in a negligible change. The PP/E approach demonstrates an enhancement in privacy gain from 67% to 83%. Concurrently, there is an augmentation in energy from 0.12 to 0.197 watts, and a concomitant escalation in time from 0.719 to 1.06, which is the threshold for time constraints. Therefore, the best approach is determined as PP/E according to privacy gain (%), energy, and time results until the task number is 500 in the topology.

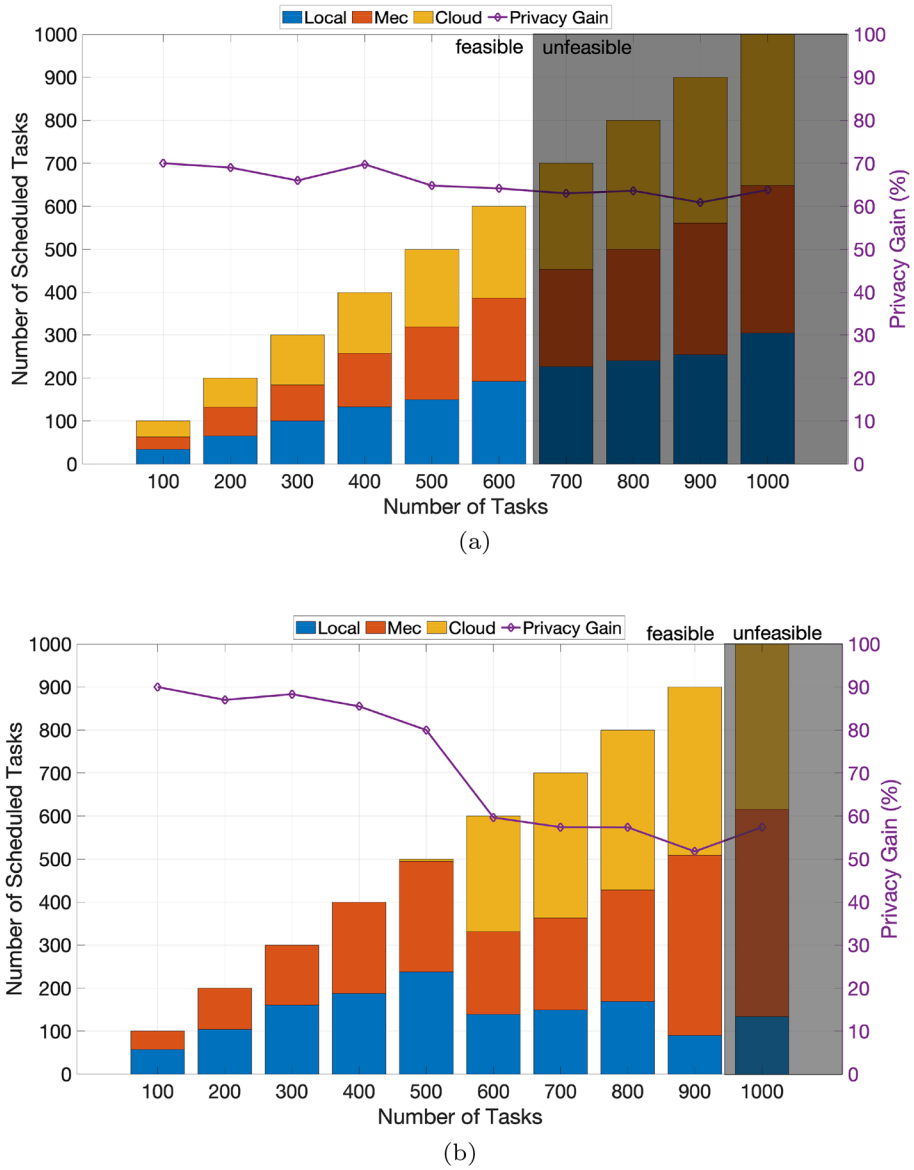

Fig. 6 Task scheduling and Privacy Gain (%) results for the proposed approach when the task number increases with (a) the conventional approach (b) the proposed approach

Table 2 Digital twin processing time for the proposed task scheduling approach when there are 500 tasks

Chromosome	2	50	100	500	1000
Iteration/	**Processing Time (min)**				
1	0.002	0.004	0.014	0.018	0.029
100	0.003	0.011	0.016	0.041	0.084
500	0.006	0.023	0.041	0.176	0.341
1000	0.007	0.041	0.067	0.311	0.675
100000	0.0141	0.305	0.582	2.88	6.209
Iteration/	**Privacy Gain (%)**				
1	65.4	65.2	65.4	68.0	65.4
100	68.4	68.0	63.8	68.8	66.0
500	63.2	73.6	72.2	72.4	73.2
1000	72.0	70.4	70.8	76.2	73.8
10000	74.0	82.6	83.6	84.0	82.6
Iteration/	**Energy (watt)**				
1	0.121	0.116	0.115	0.121	0.113
100	0.131	0.125	0.115	0.124	0.118
500	0.124	0.147	0.142	0.141	0.142
1000	0.138	0.145	0.147	0.157	0.151
10000	0.152	0.198	0.197	0.199	0.193

In Figs. 6a and 6b show the sustainability of the proposed algorithm while comparing the conventional approach. Each graph displays the total scheduled tasks in local devices, MEC, and cloud servers, respectively, as indicated by the stacked bar. The findings indicate a substantial enhancement in the sustainability of the digital twin, with a documented increase of 30%. In fact, the proposed energy- and privacy-aware genetic algorithm-based task scheduling can be implemented until 900 tasks are present in the topology. In contrast, the conventional approach is only capable of handling up to 600 tasks. When the task number is under 600, the proposed algorithm outputs approximately 18% better data privacy gain under the feasible region. When the digital twin layer is under stressed after the task number is 600, there is an acceptable loss of data privacy gain, 5% while comparing it with the conventional one. On the opposite way, if there is a chance to have an extra budget to enhance the computing capability of the digital twin or expand the time restriction of the digital twin, the data privacy gain will be much more than the current one.

In Table 2, the processing time and optimal outputs of the proposed algorithm have been analyzed with different iterations (I) and chromosome numbers (V). As the chromosome and iteration numbers increase, the processing time also exponentially rises. If the iteration number is 10,000 and the chromosome number is increased to 500, the outputs will be in unfeasible region because of the time constraint. According to privacy gain and energy results, the optimal solution is found by 10,000 iterations and 100 chromosomes in the feasible region. It results in 83% privacy gain and 0.197 watts energy consumption in the topology with 0.58 second response time. Increasing the number of iterations has a much more positive effect than increasing the number of chromosomes. For example, when the iteration number is 1000, the privacy gain (%) increases from 72% to 73.8% with an increase in the number of chromosomes. Conversely, increasing the iteration number from 1 to 10,000 with a chromosome number of 500 yields a privacy gain result of 68.0% to 84.0%. Although the number of chromosomes and iterations is increased, the gain in optimal results is a small difference. Namely, spending much more time after this threshold does not make sense for the obtained results. Therefore, these are determined optimal parameters for the proposed genetic algorithm.

6 Interpretation of the results

The smart systems discussed in this paper are built using a large number of IoT devices that execute tasks with different levels of data sensitivity. To avoid handling all tasks on local IoT devices, tasks are offloaded to MEC and cloud servers while taking time and privacy metrics into account. However, executing a large number of tasks on a MEC server can lead to scalability problems at the edge. In light of this, the time and energy performances of the conventional and proposed approaches are compared as the number of tasks executed in the system increases in the previous section. According to the summary of the results, the conventional approaches can handle up to 400 tasks in the system, whereas the proposed algorithm uses a fitness function that prioritizes energy efficiency and privacy gain (PP/E) up to a task number of 600. Specifically, it can increase scalability by a factor of 1.5. After this threshold, when the digital twin layer becomes overloaded with more than 600 tasks, it still produces nearly the same data privacy gain results as the conventional approach, where the time restriction can not be handled. To handle this restriction, thanks to digital twin monitoring and the configuration feature of the system, vCPUs can be added to enhance the computing capability of MEC servers [40, 41]. Specifically, the processing time can be reduced to an acceptable level while executing more than 600 tasks, which will result in a much greater data privacy gain than the current level.

7 Conclusion

As computation tasks are increasingly offloaded to the network edge, ensuring data privacy has become a critical concern. Task scheduling strategies should not only optimize performance metrics such as latency and energy consumption but also account for the sensitivity of tasks to prevent unauthorized access and leakage. To address this challenge,

we analyzed data leakage risks in digital twin-enabled edge networks. We proposed a sensitivity-aware and energy-efficient data leakage prevention framework that leverages digital twin technology to support secure and content-aware task offloading decisions. We modeled energy consumption and task processing time using the M/M/1 queuing model, and introduced an energy and privacy-aware genetic algorithm-based task scheduling algorithm that dynamically offloads tasks based on their sensitivity, system conditions, and latency constraints. Simulation results demonstrated that our framework outperforms conventional methods by solving the trade-off between energy efficiency and data privacy gain. By prioritizing data privacy gain while offering an energy and time-efficient implementation, our framework presents a practical and scalable approach to secure computation in edge–cloud infrastructure. As future work, we will investigate the integration of federated learning to enhance decentralized privacy preservation, addressing the challenge of maintaining privacy in a distributed environment. Specifically, ensuring data privacy while training models across distributed devices requires developing robust algorithms [42]. Additionally, we will extend our model to handle more complex task dependencies and heterogeneous network conditions. This is because tasks can also be interdependent, meaning that the scheduling of one task may affect the execution of others. Designing an efficient task scheduling framework that can accommodate such dependencies and consider privacy and energy constraints will be a complex problem.

Acknowledgements This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) 1515 Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab. Project 5239903.

Author contributions M.EÖ and E.BA have been equally contributed the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) 1515 Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab. (Grant numbers 5239903)

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

Ethics declaration Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Bozkaya-Aras, E., Erel-Özçevik, M.: Markov chain-based data privacy prioritizing for task scheduling in digital twin edge networks. In: 2024 6th International Conference on Blockchain Computing and Applications (BCCA), pp. 236–243 (2024). https://doi.org/10.1109/BCCA62388.2024.10844415
2. Bozkaya, E., Bilen, T., Erel-Özçevik, M., Özçevik, Y.: Energy-aware task scheduling for digital twin edge networks in 6g. In: 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), pp. 1–6 (2023). https://doi.org/10.1109/SmartNets58706.2023.10215892
3. ITU: Security requirements for iot service platform, series x: Data networks, open system communiactions and security, secure applications and services (2) – internet of things (iot) security. Technical report, Telecommunication Standardization Sector of ITU (January 2022)
4. Pal, S., Hitchens, M., Rabehaja, T., Mukhopadhyay, S.: Security requirements for the internet of things: A systematic approach. Sensors **20**(20) (2020) https://doi.org/10.3390/s20205897
5. Telçeken, M., Bozkaya-Aras, E.: A trustworthy federated learning model: Client selection for iot edge networks. In: 2024 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1–6 (2024). https://doi.org/10.1109/ASYU62119.2024.10756968
6. Bolat-Akça, B., Bozkaya, E.: Software-defined intrusion detection system for ddos attacks in iot edge networks. In: 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 0672–0677 (2023). https://doi.org/10.1109/DASC/PiCom/CBDCom/Cy59711.2023.10361494
7. Regulation, G.D.P.: Regulation (eu) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec. Technical report, EU Council (April (2016)
8. Guidelines, D.b.r.: Iso/iec 27400/27402 cybersecurity. Technical report, IoT security and privacy (2023)
9. Zekiye, A., Bankaoglu, O.S., Bouachir, O., Özkasap, O., Aloqaily, M.: Analysis of blockchain assisted energy sharing algorithms with realistic data across microgrids. In: 2024 6th International Conference on Blockchain Computing and Applications (BCCA), pp. 310–317 (2024). https://doi.org/10.1109/BCCA62388.2024.10844382
10. Bolat-Akça, B., Bozkaya-Aras, E.: Digital twin-assisted intelligent anomaly detection system for internet of things. Ad Hoc Networks 158, 103484 (2024) https://doi.org/10.1016/j.adhoc.2024.103484
11. Bozkaya-Aras, E.: Digital twin-enabled age of information-aware scheduling for industrial iot edge networks. Pervasive and Mobile

Computing 112, 102083 (2025) https://doi.org/10.1016/j.pmcj.2025.102083
12. Bozkaya, E., Erel-Özçevik, M., Bilen, T., Özçevik, Y.: Proof of evaluation-based energy and delay aware computation offloading for digital twin edge network. Ad Hoc Networks 149, 103254 (2023) https://doi.org/10.1016/j.adhoc.2023.103254
13. Xu, J., Yao, J.: Exploiting physical-layer security for multiuser multicarrier computation offloading. IEEE Wireless Communications Letters **8**(1), 9–12 (2019). https://doi.org/10.1109/LWC.2018.2845882
14. Bozkaya-Aras, E.: Blockchain-based secure content caching and computation for edge computing. IEEE Access 12, 47619–47629 (2024) https://doi.org/10.1109/ACCESS.2024.3383474
15. Song, Z., Ma, H., Zhang, R., Xu, W., Li, J.: Everything under control: Secure data sharing mechanism for cloud-edge computing. IEEE Transactions on Information Forensics and Security 18, 2234–2249 (2023) https://doi.org/10.1109/TIFS.2023.3266164
16. Wu, M., Li, K., Qian, L., Wu, Y., Lee, I.: Secure computation offloading and service caching in mobile edge computing networks. IEEE Commun. Lett. **28**(2), 432–436 (2024). https://doi.org/10.1109/LCOMM.2023.3347218
17. Okegbile, S.D., Gao, H., Talabi, O., Cai, J., Yi, C., Niyato, D., Shen, X.: Fles: A federated learning-enhanced semantic communication framework for mobile aigc-driven human digital twins. IEEE Network **39**(5), 238–246 (2025). https://doi.org/10.1109/MNET.2025.3526556
18. Chen, R., Yi, C., Zhou, F., Kang, J., Wu, Y., Niyato, D.: Federated digital twin construction via distributed sensing: A game-theoretic online optimization with overlapping coalitions. IEEE Trans. Mob. Comput. **24**(11), 12221–12238 (2025). https://doi.org/10.1109/TMC.2025.3582755
19. Li, Y., Zeng, D., Gu, L., Zhu, A., Chen, Q., Yu, S.: Pasto: Enabling secure and efficient task offloading in trustzone-enabled edge clouds. IEEE Trans. Veh. Technol. **72**(6), 8234–8238 (2023). https://doi.org/10.1109/TVT.2023.3237204
20. Roshan, R., Matam, R., Mukherjee, M., Lloret, J., Tripathy, S.: A secure task-offloading framework for cooperative fog computing environment. In: GLOBECOM 2020 - 2020 IEEE Global Communications Conference, pp. 1–6 (2020). https://doi.org/10.1109/GLOBECOM42002.2020.9322509
21. Wu, M., Song, Q., Guo, L., Lee, I.: Energy-efficient secure computation offloading in wireless powered mobile edge computing systems. IEEE Trans. Veh. Technol. **72**(5), 6907–6912 (2023). https://doi.org/10.1109/TVT.2023.3236327
22. Yang, N., Fan, X., Puthal, D., He, X., Nanda, P., Guo, S.: A novel collaborative task offloading scheme for secure and sustainable mobile cloudlet networks. IEEE Access 6, 44175–44189 (2018) https://doi.org/10.1109/ACCESS.2018.2853634
23. He, X., Jin, R., Dai, H.: Physical-layer assisted secure offloading in mobile-edge computing. IEEE Trans. Wireless Commun. **19**(6), 4054–4066 (2020). https://doi.org/10.1109/TWC.2020.2979456
24. Xiang, H., Yi, C., Wu, K., Chen, J., Cai, J., Niyato, D., Shen, X.: Realizing immersive communications in human digital twin by edge computing empowered tactile internet: Visions and case study. IEEE Network **39**(4), 271–279 (2025). https://doi.org/10.1109/MNET.2024.3507157
25. Chen, J., Shi, Y., Yi, C., Du, H., Kang, J., Niyato, D.: Generative-ai-driven human digital twin in iot healthcare: A comprehensive survey. IEEE Internet Things J. **11**(21), 34749–34773 (2024). https://doi.org/10.1109/JIOT.2024.3421918
26. Mao, B., Liu, J., Wu, Y., Kato, N.: Security and privacy on 6g network edge: A survey. IEEE Communications Surveys & Tutorials **25**(2), 1095–1127 (2023). https://doi.org/10.1109/COMST.2023.3244674
27. Liao, Z., Pang, X., Zhang, J., Xiong, B., Wang, J.: Blockchain on security and forensics management in edge computing for iot: A comprehensive survey. IEEE Trans. Netw. Serv. Manage. **19**(2), 1159–1175 (2022). https://doi.org/10.1109/TNSM.2021.3122147
28. Lyu, Z., Cheng, C., Lv, H., Song, H.: Blockchain based intelligent resource management in distributed digital twins cloud. IEEE Network **38**(4), 143–150 (2024). https://doi.org/10.1109/MNET.2023.3326099
29. Xu, C., Zhang, P., Xia, X., Kong, L., Zeng, P., Yu, H.: Digital-twin-assisted intelligent secure task offloading and caching in blockchain-based vehicular edge computing networks. IEEE Internet Things J. **12**(4), 4128–4143 (2025). https://doi.org/10.1109/JIOT.2024.3482870
30. Yigit, Y., Maglaras, L.A., Buchanan, W.J., Canberk, B., Shin, H., Duong, T.Q.: Ai-enhanced digital twin framework for cyber-resilient 6g internet of vehicles networks. IEEE Internet Things J. **11**(22), 36168–36181 (2024). https://doi.org/10.1109/JIOT.2024.3455089
31. Erel-Özçevik, M., Bozkaya-Aras, E.: Digital twin-empowered service provisioning framework for cloud-assisted internet of things. In: 2025 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1092–1097 (2025). https://doi.org/10.1109/ICCWorkshops67674.2025.11162421
32. Bozkaya-Aras, E.: Optimizing service migration in iot edge networks: Digital twin-based computation and energy-efficient approach. In: 2025 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2025). https://doi.org/10.1109/WCNC61545.2025.10978782
33. Yang, Y., Shi, Y., Yi, C., Cai, J., Kang, J., Niyato, D., Shen, X.: Dynamic human digital twin deployment at the edge for task execution: A two-timescale accuracy-aware online optimization. IEEE Trans. Mob. Comput. **23**(12), 12262–12279 (2024). https://doi.org/10.1109/TMC.2024.3406607
34. Bozkaya-Aras, E., Onel, T., Eriskin, L., Karatas, M.: Intelligent human activity recognition for healthcare digital twin. Internet of Things 30, 101497 (2025) https://doi.org/10.1016/j.iot.2025.101497
35. Bozkaya, E.: A digital twin framework for edge server placement in mobile edge computing. In: 2023 4th International Informatics and Software Engineering Conference (IISEC), pp. 1–6 (2023). https://doi.org/10.1109/IISEC59749.2023.10390994
36. Bozkaya, E., Canberk, B., Schmid, S.: Digital twin-empowered resource allocation for 6g-enabled massive iot. In: 2023 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 727–732 (2023). https://doi.org/10.1109/ICCWorkshops57953.2023.10283649
37. Bolat-Akça, B., Bozkaya-Aras, E., Canberk, B., Buchanan, B., Schmid, S.: An intelligent digital twin model for attack detection in zero-touch 6g networks. In: 2024 IEEE International Conference on Communications Workshops (ICC Workshops) (2024)
38. Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M.: Fundamentals of Queueing Theory, 4th edn. Wiley-Interscience, New York, NY, USA (2008)
39. Kiran, N., Pan, C., Wang, S., Yin, C.: Joint resource allocation and computation offloading in mobile edge computing for sdn based wireless networks. Journal of Communications and Networks **22**(1), 1–11 (2020). https://doi.org/10.1109/JCN.2019.000046

40. Sabella, D., Nikaein, N., Huang, A., Xhembulla, J., Malnati, G., Scarpina, S.: A hierarchical mec architecture: Experimenting the raven use-case. In: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), pp. 1–5 (2018). https://doi.org/10.1109/VTCSpring.2018.8417826
41. Spatharakis, D., Dimolitsas, I., Dechouniotis, D., Papathanail, G., Fotoglou, I., Papadimitriou, P., Papavassiliou, S.: A scalable edge computing architecture enabling smart offloading for location based services. Pervasive and Mobile Computing 67, 101217 (2020) https://doi.org/10.1016/j.pmcj.2020.101217
42. Bolat-Akça, B., Telçeken, N.M., Bozkaya-Aras, E.: Digital twin-enabled federated misbehavior detection model for internet of vehicles networks. In: 2025 IEEE 36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–6. (2025) https://doi.org/10.1109/PIMRC62392.2025.11275116

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.