Application of Edge Computing-Based Information-Centric Networking in Smart Cities

Munqith Saleem[1,*], Hanan Burhan Saadon[2], Marwa S. Mahdi Hussin[3], Tamarah Alaa Diame[4], Raaid Alubady[5], Mohd K. Abd Ghani[6], Hatıra Günerhan[7]

[1]Medical instruments engineering techniques, Al-farahidi University, Baghdad, Iraq:
[2]Department of Medical Devices Engineering Technologies, National University of Science and Technology, Dhi Qar, Nasiriyah, Iraq
[3]Computer Technologies Engineering, Al-Turath University College, Baghdad,Iraq
[4]Technical Computer Engineering Department, Al-Kunooze University College, Basrah, Iraq
[5]Department of Computer Technical Engineering, Technical Engineering College, Al-Ayen University, Thi-Qar, Iraq
[6]Biomedical Computing and Engineering Technologies (BIOCORE) Applied Research Group, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal 76100, Malaysia
[7]Department of Mathematics, Faculty of Education, Kafkas University, Kars, Turkey

Emails: Munqith Saleem@uoalfarahidi.edu.iq; hanan.saadon@nust.edu.iq; marwa.saad@turath.edu.iq; Tamarah. Alaa@Kunoozu. Edu. Iq; raaidalobady@alkitab.edu.iq; khanapi@utem.edu.my; hatira.gunerhan@kafkas.edu.tr
*Corresponding Author: Munqith Saleem@uoalfarahidi.edu.iq

Abstract

Many data resources and network availability are needed for the smart city applications to execute at their highest efficiency level Many interconnected devices in a smart city produce vast quantities of data and will likely have new uses in the near future. In smart cities, the Internet of Things (IoT) and 5G beyond networks provide dependable, large-scale data exchange and communication. A new intelligent ecosystem is the goal of 5G, and the technology that will make it possible is the next-gen networking technologies. The drawback of smart devices is their limited computational capability. Adding in-network caching into information-centric edge networks allows them to overcome this obstacle. Hence, this study suggests an Adaptive Information-Centric Network based on Edge Computing Framework (AICN-ECF) to reduce data traffic and latency with high security in smart cities. Integrating EC and ICN allows content distribution to be handled quickly, improving user experience. This study provides an ICN-based edge caching system with four cache attributes for managing large multimedia data traffic in smart cities built on the Internet of Things. At the base station (BS) application layer, there is support for ICN and device-to-device (D2D) communication, which allows for caching of requested material at the network's edge. This layered design is the first step in the process. Secondly, to facilitate efficient caching, a selection has been offered to cache contents at network nodes in a layered network design, considering a variety of centrality indicators. Finally, this study provides a method for caching material close to the delivery path in ICN network layers, allowing for rapid content distribution by using near-path caching. The experimental findings demonstrate that the suggested AICN-ECF model increases the cache hit ratio of 98.7%, content retrieval time of 97.8%, data security ratio of 96.5%, data transmission ratio of 95.6% and delay ratio of 11.2% compared to other popular models.

Keywords: Edge computing; networking; information access; smart city; memory

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

1. Introduction

Smart cities enable people to experience secure, reliable, and sustainable developments using Information and Communication Technology (ICT) architecture to relate to smart cities. Smart city services refer to various processes that result in innovative cities [1]. Network performance, dependability, and security have become more important concerns due to the explosion of smart cities and IoT applications [1]. Problems with service availability, dependability, sustainability, and security have emerged from the distributed nature of the IoT and smart city infrastructure [2]. Applications like cognitive support, smart health, transportation systems, and social services may all benefit from the decentralized processing and communication infrastructure that edge computing provides [3]. Applications that rely on real-time processing may benefit from edge computing, which boosts speed and decreases end-to-end latency [4]. The transmission overhead on the network is becoming more heavy as the number of edge devices continues to rise [5]. However, the growing need for edge networks is beyond the capabilities of conventional, centralized cloud computing [6]. On the other hand, computing at the network's periphery could reduce pressure, and the distributed computing power and storage space at its edges can substantially enhance the capacity to handle tasks [7]. With edge computing, some of the memory and processing power is relocated from the data centre to the very network edge, where it is most required [8]. Because of this, there would be a lot of advantages, as a lot less data would need to be sent to the central clouds, and a lot more data could be handled on the edge [9]. If massive data growth is going to be necessary in the near future, this new computer paradigm is an appropriate match [10].

The information-centric networking (ICN) architecture is a great idea for the future of the Internet since it allows for more efficient content forwarding by storing data in routers [11]. One advantage of ICN over other caching mechanisms is that it installs cache at the network layer, which usually has less overhead than at the application layer. [12]. Since content delivery networks (CDNs) depend on deploying proxy servers, this makes them less beneficial than this network caching approach. [13]. Second, applications do not detect or interfere with the network layer caching method. Deploying caching on the edge networks via ICN is a potential alternative [14]. ICN is another emerging paradigm based on content-centric networking, often called future internet architecture. It is similar to EC yet takes a different approach [15]. When it comes to receiver-driven communication, ICN makes name-based content routing possible. For future networks, ICN's in-network caching function may handle heavy traffic loads and significant delays [16]. ICN has the potential to facilitate EC by empowering end users via communication. Before EC, every request from users had to make its way to the network's core to get the material they needed [17]. Host-based communication is required for the requests to reach their destination [18]. Due to its content name-based communication, ICN makes it easy to control the difficulties that arise during IP-based communication [19]. Using a naming strategy to address the information-dissemination challenge, ICN represents a potential future paradigm for the Internet. Unlike the traditional Internet, which uses IP addresses to obtain content, this method relies on the content's name. [20].

The major contribution of the paper is

- Developing the Adaptive Information-Centric Network based on Edge Computing Framework (AICN-ECF) to reduce data traffic and latency with high security in smart cities
- Introducing the ICN with an edge-based caching strategy to release redundant traffic and decrease delay.
- The experimental outcomes have been performed, and the suggested AICN-ECF enhances the cache hit ratio, content retrieval time, data security, and delay ratio compared to existing approaches.

The rest of the paper is planned as follows: section 2 deliberates the related works, section 3 proposes the AICN-ECF model, section 4 discusses the numerical findings, and Section 5 concludes the research paper.

2. Related Study

Ning Chen et al. [21] proposed the Edge Intelligent Networking Optimization (EINO) for IoT in smart cities. The suggested framework efficiently uses computing resources while reducing training time and costs using multi-core CPU and smart edge approaches. Along with optimizing the network design to use smart terminal nodes' contributions, the best-performing framework analyzes the dispersed communication model of EC. To prove that the framework is effective in different topologies, the author compares it to other cutting-edge algorithms that aim to strengthen the IoT architecture in smart cities.

Khalid Haseeb et al. [22] proposed the intelligent and secure edge-enabled computing (ISEC) model for sustainable cities utilizing Green IoT to reduce liability in energy management and information security for information transportation. This model seeks to build a communication strategy. The suggested technique trains sensors to anticipate the best pathways to the edge server by generating optimum characteristics for data routing using deep learning. Combining distributed hashing with chaining makes security solutions using an effective computing system easier. The numerical findings show that the suggested ISEC model outperforms competing

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

alternatives in terms of energy usage (21%), network throughput (15%), end-to-end latency (12%), network overhead (52%) and route interruption (36%).

Divya Gupta et al. [23] discussed the ICN-IoT content caching technique to facilitate heterogeneous IoT architecture, which allowed for collaborative filtering inside the edge cloud. Intelligent content caching on edge nodes for traffic control at the cloud database is the goal of this collaborative filtering-based approach. According to the data, the suggested approach outperforms the best-considered Leave Copy Down (LCD) by 12% in content retrieval time, an average of 15% in cache hit ratio, and 28% in average hop count. Based on the research, the author is optimistic that the suggested approach will provide useful results for future studies.

Kai Peng et al. [24] discussed the multi-objective collaborative optimization for the smart city (MOCOSC) based on Mobile edge computing (MEC). The author considers the possibility of simultaneously optimizing MEC-enabled smart cities' mobile devices and edge servers to boost the system's overall performance. Reducing energy and time consumption, maintaining edge servers' load balancing, increasing average resource utilization, and satisfying deadline constraints of delay-sensitive applications are all achieved via the technical implementation of a new multi-objective compute offloading method. The suggested method was adequately tested to demonstrate its efficacy and superiority in many contexts.

Laha Ale et al. [25] recommended Spatio-Temporal Bayesian Learning (STBL) for MEC resource planning in smart cities. Using a MEC server at an ideal location and assigning MEC resources are critical for successfully providing service needs in a smart city, especially when service demands display spatial-temporal patterns. To that end, it is critical to understand how resource demands are distributed over time and space. To help resource management and MEC deployment, the author first suggests an STBL to understand and anticipate how MEC resource demand is distributed across space and time. In addition, the findings show that the suggested strategy may reach extremely high accuracy when trained and evaluated on real-world data. Third, by modelling task offloading, the author shows how to implement the suggested approach. The simulation findings demonstrate that allocating resources according to the models' predictions is more efficient than distributing resources evenly across all servers in unknown locations.

Pengjie Zeng et al. [26] presented the Trust-Based Multi-Agent Imitation Learning (T-MAIL) for green energy computing in smart cities. This article begins with the author's complete task-offloading incentive models, allowing edge devices to take advantage of the full benefits of task re-offloading and local processing. The second aspect is that the author suggested an active trust acquisition approach to correctly and efficiently acquire the device's trust. Ultimately, multi-agent imitation learning incorporates the novel trust acquisition mechanism and task offloading reward scheme. The typical time to complete a task has been reduced by 52.5% and 27.5% compared to current task offloading solutions approaches. With this technique, the task offloading rate is 19.2%, and the trust difference ratio between reliable and unauthorized devices can extend to 56.1%.

Manvitha Gali and Aditya Mahamkali [27] introduced the Distributed Deep Meta learning-driven Task Offloading (DDMTO) for Smart City IoT with Edge-Cloud Computing. Improved portability and efficient use of Deep Neural Networks (DNN) for offloading decision-making were the goals of this research. A network's output is the result of providing data from the BP algorithm's hidden layers. Inputs are contrasted with the intended outputs, and differences are used to monitor problems as they spread from hidden levels to input layers and then back again. Different neuron weights are affected by the restoration of flows. During an epoch, the inputs are transformed into outputs, and the outputs are transformed back into inputs.

Xiaolong Xu et al. [28] offered edge content searching with a deep spatiotemporal residual network (DSRN) for the Internet of Vehicles (IoV) in smart cities. E-Cache is suggested for edge content caching in smart cities that incorporate service need prediction and deliver high-quality services while assuring resource efficacy with edge content caching. After that, ST-ResNet and DSRN were used to forecast the vehicles' future servicing needs. The next step is to develop initial content caching methods using the expected service demands; these are then fine-tuned using a multi-objective optimization that seeks to reduce the vehicle services' execution period and energy usage. The efficacy and efficiency of E-Cache with spatiotemporal traffic route large information was shown in the end by experimental evaluations.

Lukman Adewale Ajao and Simon Tooswem Apeh [29] proposed the Petri net and genetic algorithms-based reinforcement learning for Secure edge computing susceptibilities in smart cities. A distributed authorization mechanism is initially suggested as a common trust model to handle network information outflows. This method is used on a Petri Net-based safe framework known as secure trust-aware philosopher privacy and authentication (STAPPA) to prevent network privacy breaches. During agent learning in the environment, GAL (Genetic Algorithm-based Reinforcement Learning) is used to optimize search, identify anomalies, and find the shortest path. In a simulated network setting, the detection and accuracy rates achieved by a reinforcement learning-based secure framework were 98.75, 99, 99.50, 99.75, and 100%, respectively.

Weixi Wang et al. [30] suggested Edge Intelligent Federation Learning (EIFL) for data processing of traffic digital twin (DT) in smart cities. A smart traffic perception system based on EC paired with DT is built using the latest technologies in edge computing. Additionally, the Single Shot MultiBox Detectors (SSD) method is enhanced by residual networks derived from the SSD-ResNet50 technique, and DarkNet-53 is enhanced to boost the accuracy

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

of traffic scene detection. As a last step, tests are carried out to confirm the results of the data improvement approach and the modified algorithms. High recognition accuracy, fast training speed, and a beneficial training effect are shown by the experimental findings of the SSD-ResNet50 and the upgraded DarkNet-53 algorithms. The SSD-ResNet50 method and the enhanced DarkNet-53 algorithm both have a 4.25ms and a 6.37ms reduction in recognition time compared to the original algorithms.

Based on the investigation, there are several issues in existing methods in attaining a high cache hit ratio, content retrieval time, data security, and reduced delay ratio. Hence, this paper suggests an Adaptive Information-Centric Network based on Edge Computing Framework (AICN-ECF) to reduce data traffic and latency with high security in smart cities.

3. Adaptive Information-Centric Network based on Edge Computing Framework (AICN-ECF)

Scalable content distribution, security, mobility, and other networking-related issues have recently come to light due to the Internet's rapid expansion and the proliferation of related applications. The premise that communication is more often used to distribute data than to link two end hosts has given rise to the information-centric network (ICN) as a potential future network design. ICN allows mobility support, packet-level security, in-network caching, name-based routing independent of topology, and multicast/anycast. With its push-based (publish/subscribe) and pull-based (interest/data) communication mechanisms, ICN is well-suited to distribute information in smart city services, allowing for effective and timely data supply to users. Problems arise when trying to handle large volumes of IoT data effectively. In addition, rather than focusing on connecting various devices, IoT applications aim to extract pertinent data. Because of this, the information-centric demands of IoT applications conflict with the present Internet's host-centric architecture. 5G networks aim to significantly decrease latency and bandwidth constraints while delivering improved dependability; one architectural paradigm that has recently emerged, Edge Computing (EC), does this by bringing computing closer to end-users. Through EC, computation may be transferred from end devices to edge nodes. Hence, this paper suggests Adaptive Information-Centric Networking based on Edge Computing Framework (AICN-ECF) to reduce data traffic and latency with high security in smart cities.

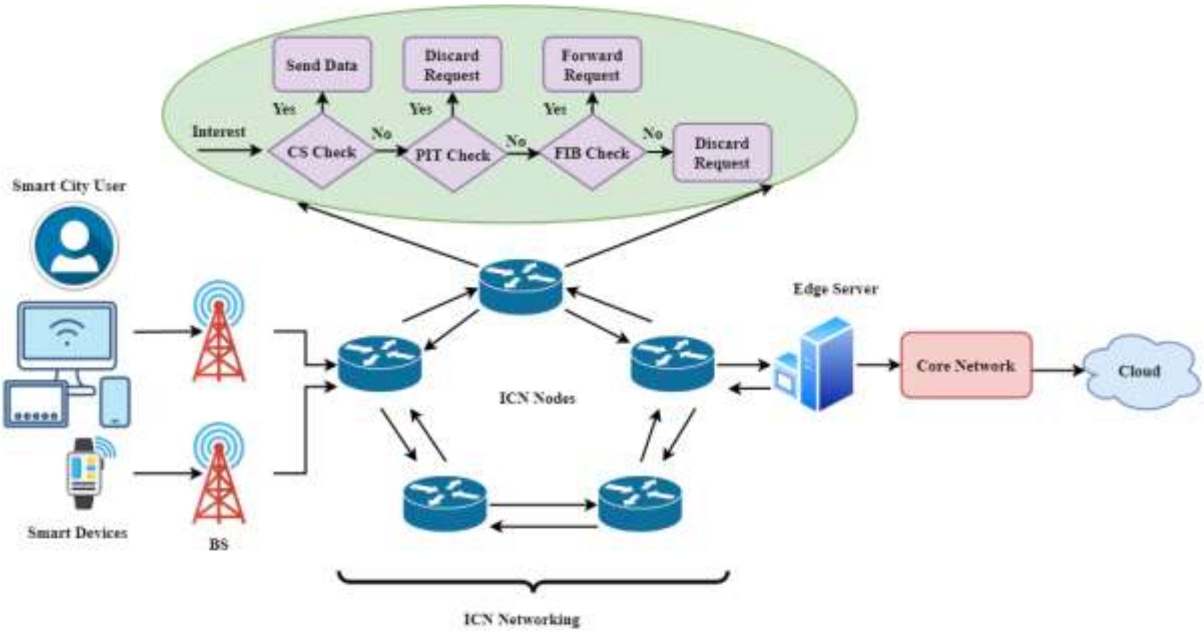

Figure 1: Proposed AICN-ECF model

Figure 1 shows the proposed AICN-ECF model. The network's central node gets overloaded due to the massive traffic generated by data in smart cities. The initial step in implementing ICN in smart cities is to deploy it at the end-user device layers to facilitate D2D communication. This will allow communication inside the smart city while reducing traffic at the core network. One advantage of Device-to-Device (D2D) communication is that it eliminates the need to transport requests for cached data to base stations or access points, allowing for easier local processing. Second, an ICN network is separated into a multi-tiered network hierarchy at the Base Station (BS) application layer. Users cluster the dispersed network nodes once they have made a content request. Nodes in each cluster would originate from different levels of the network, according to the topology. Nodes farther away from the edge devices are not considered when calculating the provided network latency for content fetching. As a result, the topology radius determines the maximum number of tiers that may exist in the ICN network. The core network receives content requests directly if they cannot be handled by any level of the ICN network structure. Nodes in the top layer are ordered by centrality, with the more important ones caching responses from the

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

originating server. A key benefit of customizable clusters is their ability to adapt their layout to different network topologies. On each node of the network, ICN has implemented three data structures: the Pending Interest Tables (PIT), which keeps track of the interface that has received interest packet with similar name prefixes (requesting similar content); the Forwarding Information Base (FIB), which keeps track of the interface that can send out interest packets to fetch contents, and the Content Stores (CS), which delivers a facility to store named-content. The CS in the ICN node can store the designated content indefinitely, unlike the IP routers' ephemeral buffers discarded after packet forwarding. That is to say, every node in the CCN network with named content in its cache may act as an additional source for serving such content because named content is not location-dependent. CS in ICN deliberately keeps the most demanding material to obtain a high cache hit rate. In automation, a closed ecosystem uses edge computers to handle data processing and make decisions locally rather than depending on an inaccessible cloud. ICN further improves content caching and decouples the receiver and sender to provide smart services effectively. EC may reduce the load on the core network and enable low-delay access by caching popular and frequently accessed items at the network's edge. This study executes these operations while minimizing the usage of cloud bandwidth and making optimum use of the resources already available in our network. The operational difficulties of mobile networks, such as bandwidth bottlenecks, may be slightly reduced with the help of EC.

For every edge node (EN) and contents, the sum of every value of the attribute is equivalent to 1, i.e., $\sum_{j=1}^{I} \quad f_j = 1$ for each content C and edge nodes.

To compute the number of epochs a particular contents C_p has been requested by every EN_q, a history of request is formed. The values of a specific cell $Req_{q,p}$ is, thus, computed as,

$$Req_{q,p} = \frac{EN_q * C_p}{I} \leq 1 \qquad (1)$$

As shown in equation (1), Req is significant and would be beneficial for determining the content assignment inside edge nodes.

The cloud centre is an enormous databases with an ability equivalent to vol. Every content requested by IoT users can be providing via a conventional cloud information centre and with extended delays because of data traffic bottlenecks. To provide users with decreased latency, the edge node in edge clouds might cache often accessed content. This study considers homogeneous cache capability for every edge node to simplify our design approach. Therefore, this study assumes each edge node is prepared with caches of equivalent sizes, $Size_K$ is

$$Size_K = vol * \sigma \quad (2)$$

As shown in equation (2), where σ comprises values between 1 and 0, i.e., $0 < \sigma < 1$, and vol signifies the overall capability of cloud server databases. $Size_K$ indicates minor cache spaces accessible to every edge node compared to overall volume vol.

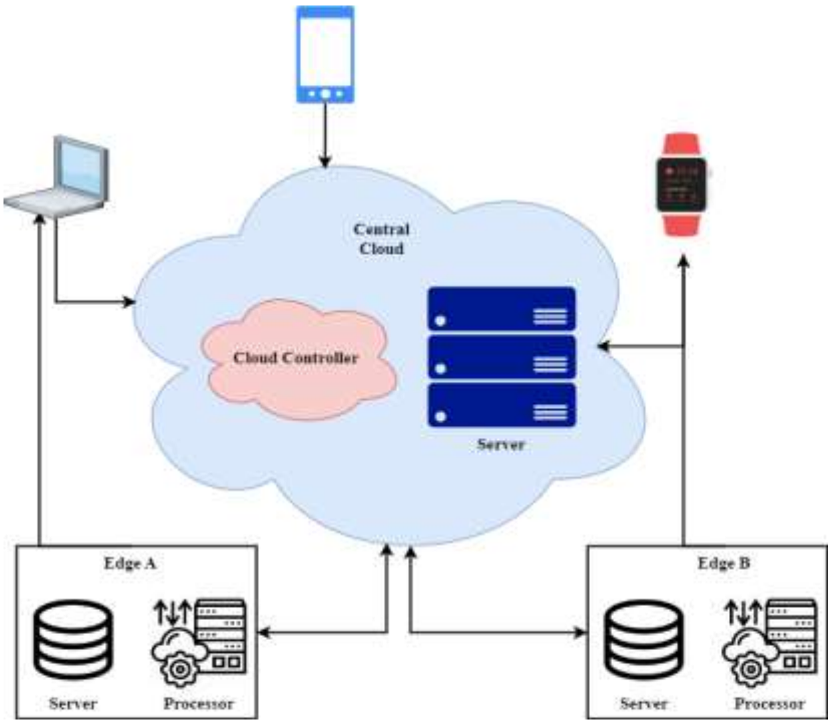

Figure 2: General Architecture of Distributed Edge Computing for Smart City Services

Figure 2 shows the General Architecture of Distributed Edge Computing for Smart City Services. Using security measures, encryption, standardization, authorization, authentication, curation, governance, and network

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

approaches, computing operations are integrated into each layer of the suggested computing architecture for smart cities. Computing on the cloud, computing at the edge, and mobile computing make up the backbone of smart city infrastructure. Data centres contain the cloud's infrastructure, as well as its physical facilities and the ability to do complicated analyses and visualizations. Services are provided by users by connecting the servers over high-speed networks. Data centres are often situated in less densely populated areas that provide a reliable power supply and are less likely to be affected by natural disasters. Through the Internet, the edge-computing platform communicates with the central cloud. They can exchange data with each other due to their dual communication capabilities. To increase processing and delivery speeds, the edge servers might combine their resources and reduce the load on the central servers. Users' mobile devices, capable of data processing and mobility, constitute the mobile-computing platform. Wireless networks are another option for transmitting data from mobile devices to central clouds. Applications requiring interactions link edge- and mobile-computing systems. Though diverse in how they work together to process the services and applications of smart cities, the three computing paradigms are interdependent and mutually supportive in the design. Instead of continuously sending data back to central servers, edge computing moves critical data processing to the network's periphery. This is in contrast to cloud computing, which necessitates that all components be linked to the central cloud. The cloud processes massive amounts of data to discover optimization solutions or back decisions. Thus, edge-enabled devices can collect and process information instantly, enabling them to react more rapidly and effectively, whereas mobile computing pertains to the rise of new interfaces and devices and can process data on mobile devices. In addition, the central cloud might handle the analytics, storage, and processing of more complicated data. Compared to central clouds, edge computing typically handles simpler data processing, storage, and forwarding. However, not all mobile devices can do anything more complex than basic data processing. By combining the ICN models, this study can overcome the inefficiency of processing and calculating massive amounts of data. Ensuring the security and safety of the entire system is the responsibility of the network connecting the edge, mobile device, and central clouds.

Every edge node is partitioned into several clusters to benefit from an intelligent content caching strategy. Based on the entry existing in Req where value may be determined utilizing Expression (1), this study applies k-means clustering to cluster the edge node into H cluster (CLS). This study uses cosine distance metrics to compute the distance among any pair of EN, which is EN_j and EN_i. EN, which belongs to similar clusters, will have a distance value roughly near 0, though EN belonging to various clusters would have distance values near 1. Furthermore, the zero value signifies the edge node in the middle of two clusters.

$$distance(EN_j, EN_i) = 1 - \frac{\sum_{n=1}^{M} Req_{j,n} Req_{i,n}}{\sqrt{\sum_{n=1}^{M} (Req_{j,n})^2}\sqrt{\sum_{n=1}^{M} (Req_{i,n})^2}} \quad (3)$$

Here in equation (3), $Req_{j,n}$ signifies the quantity of requests EN_j has prepared for content C_n. Yet, the remainder of the slice $Size_K$ is utilized to cache contents based on the maximum likelihood of future content requests. This research can use content-based collaborative filtering to estimate the likelihood of future requests for existing material. To back up intelligent caching, this research suggested a content caching method on each edge cloud that relies on collaborative filtering; this would allow for the caching of contents according to their local popularity and anticipated future needs. Next, an algorithm was developed using edge clustering and caching to fulfil the Quality of Experience (QoE) standards for user-fetched network content.

The desired IoT network computation cycle for the task t_j are signified as D_j . The computation period of tasks t_j to be executed at the user end is denoted by E_j (Equation (4)), whereas the energy consumption of the smart device is symbolized by Q_j as in Equation (5)).

$$E_j = D_j / D_v \quad (4)$$

$$Q_j = D_j (D_v)^2 \quad (5)$$

The responsibility of handling smart devices that do not meet the requirements of an assumed task is transferred to an edge server. The time needed to transmit data to the server is much less than the execution outcomes after offloading, as extensive research shows. The distance amongst the edge server and mobile user in the y and x directions is signified as y_j and x_j, correspondingly. Here, q_j signifies the transmitting energy of the smart device, ρ^2 denotes additive Gaussian noise in transmission paths, and θ signifies standard path loss propagations. Consequently, the signal-to-noise ratio (SNR) is computed utilizing Equation (6)

$$SNR_j = (q_j \times y_j \times x_j)^{-\theta/\rho^2} \quad (6)$$

If the bandwidth A is known, the smart device's transmission rates R_j, is articulated utilizing Equation (7).

$$R_j = A(1 + SNR_j) \quad (7)$$

The aggregate of the execution periods of tasks that have been earlier offloaded to edge servers are articulated by W_l as in Equation (8).

$$W_l = \sum_{\forall t_j \in N_l} TE_j^l \quad (8)$$

Additionally, the server utilization is symbolized by V_w, and the edge server's capability to which a task is heading for offload is articulated by D_w as in Equation (9).

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

$V_w = \frac{W_l}{D_w} * 100$ (9)

Lastly, the edge computation period D_h for tasks t_j is identified by Equation (10). If $R_j < D_h$ the necessary task is offloaded at edge servers.

$D_h = \frac{D_j}{D_w}$ (10)

Computation offloading in edge computing is defined by low energy consumption, latency, and server utilization to choose an appropriate edge server for a task. Users' constant movement delays the outcomes of task offloading to smart devices. Before sending data to the cloud, processing it locally might assist in aggregating or filtering it, lowering traffic volume. Secondly, intermediate nodes are often needed for data processing in IoT applications. These nodes are located between the user node and the data sources. The existing Internet's end-to-end communication mechanism is inappropriate to meet this need. This paper uses ICN as the fundamental network infrastructure to facilitate the transmission and processing of data from the IoT. The information-centric nature of IoT applications is a good match for ICN as well. Users of these applications would rather acquire content and services to link more devices.

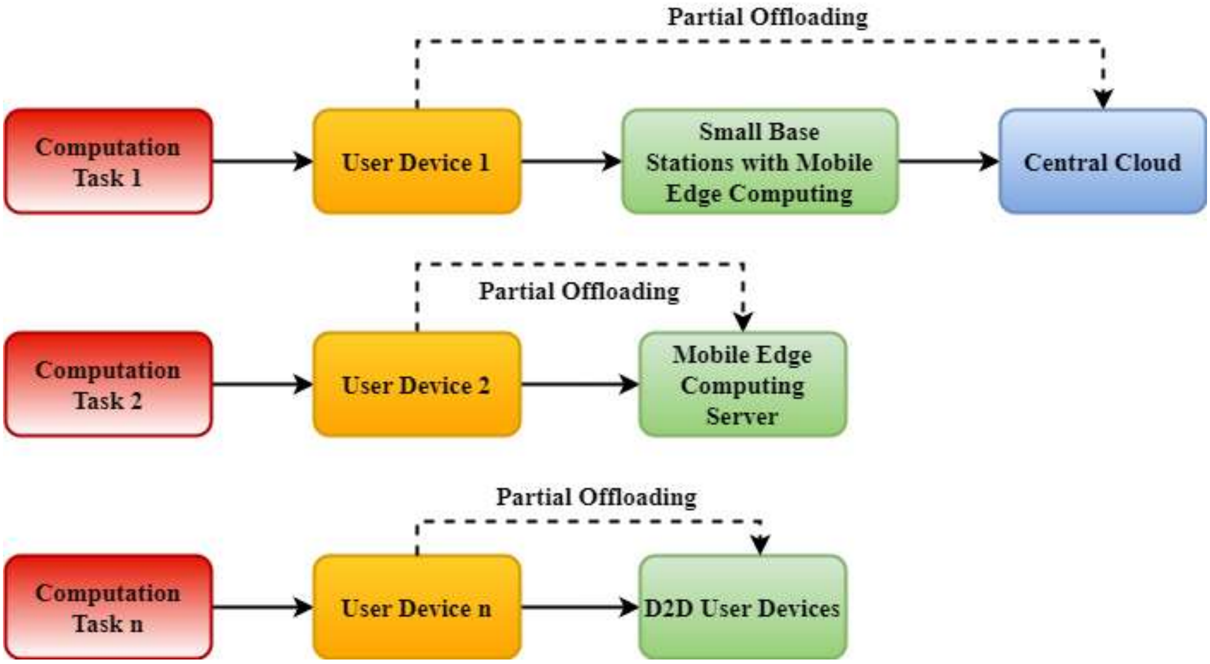

Figure 3: Representation of Computational Offloading in Edge Computing

Figure 3 shows the representation of computation offloading in edge computing. During each offloading phase, the user device can either complete their duties on the D2D networks or send them to a server in the distant cloud or mobile edge computing (MEC). Within the time frame of the request, the user's device actively participated in the discovery process by sending and receiving discovery signals at regular intervals and in sync. While a device discovery phase is in progress, user equipment (UE) actively transmitting signals might send discovery signals that other user devices can pick up on. The detection signals must include application-related and identity-related data (such as cache status). This way, the UE that is now active would be able to connect to the idle UEs that are most appropriately matched to its cache or those that are mismatched. Note that the user device servicing the Small Base Station (SBS) has complete control over the device discovery and connection establishing timeframes. Considered is a quasi-static situation in which the collection of user devices does not change throughout the compute offloading time. This study examines four different offloading modes: cloud partial offloading, cache-mismatched D2D partial offloading, MEC partial offloading, and cache-matched D2D partial offloading. By balancing server processing restrictions with energy consumption and serving latency, this research seeks to reduce system costs. Task offloading ratios, data traffic, CPU-cycle allocation in the cloud and MEC servers, and the offloading approach are all optimized together. This study design a combined optimization problem to reduce the overall offloading costs. The entire energy usage and offloading delay are multiplied linearly to get the offloading cost. Compared to existing models, the proposed AICN-ECF model increases the cache hit rate, content retrieval time, delay ratio, data security ratio, and data transmission ratio.

4. Results and Discussion

This paper presents Adaptive Information-Centric Networking based on Edge Computing Framework (AICN-ECF) to reduce data traffic and latency with high security in smart cities. The data are taken from the Edge IoT and IIoT cyber security Kaggle dataset [31]. Integrating many IoT (Internet of Things) sensors for data collection

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

and management and numerous other technological advancements in city centres will shape the future of data and automation in urban life. Smart Cities are like a customer service experience for those who live in cities. The Internet of Things data comes from various sources, including over ten IoT devices (e.g., inexpensive digital thermometers and humidity sensors, ultrasonic level detectors, soil moisture sensors, pH meters, heart rate monitors, flame detectors, etc.). Nevertheless, this study discovers and examines fourteen assaults on IIoT and IoT communication protocols. These assaults are classified into five dangers: denial-of-service/double-digital-denial, information collecting, injection, man-in-the-middle, and malware. This dataset assesses the efficacy of ML methods in federated and centralized learning modes after processing and examining the suggested realistic security dataset. The performance of the suggested AICN-ECF model has been examined based on metrics such as cache hit rate, content retrieval time, delay ratio, data security ratio and data transmission ratio.

(i) Cache Hit Ratio (CHR)

A caching strategy's effectiveness may be measured by its cache hit ratio. Accordingly, it establishes the ratio of requests fulfilled to total requests. A high CHR always indicates less load on core networks since most requests are fulfilled by middle nodes. This study examines how well the suggested CHR approach works in the first set of experiments. If similar information is found in the router's delay when checking a single cache for a specific content request, it is recorded as a hit or miss. The material is retrieved from the server whenever there is a cache miss. The amount of requests handled by cache nodes distributed throughout edge devices is CHR. Figure 4 demonstrates the cache hit rate. The formula in the following equation (11) determines the routers' efficiency.

$$CHR = \frac{Cache\ hits}{Cache\ hits + Cache\ miss} \qquad (11)$$

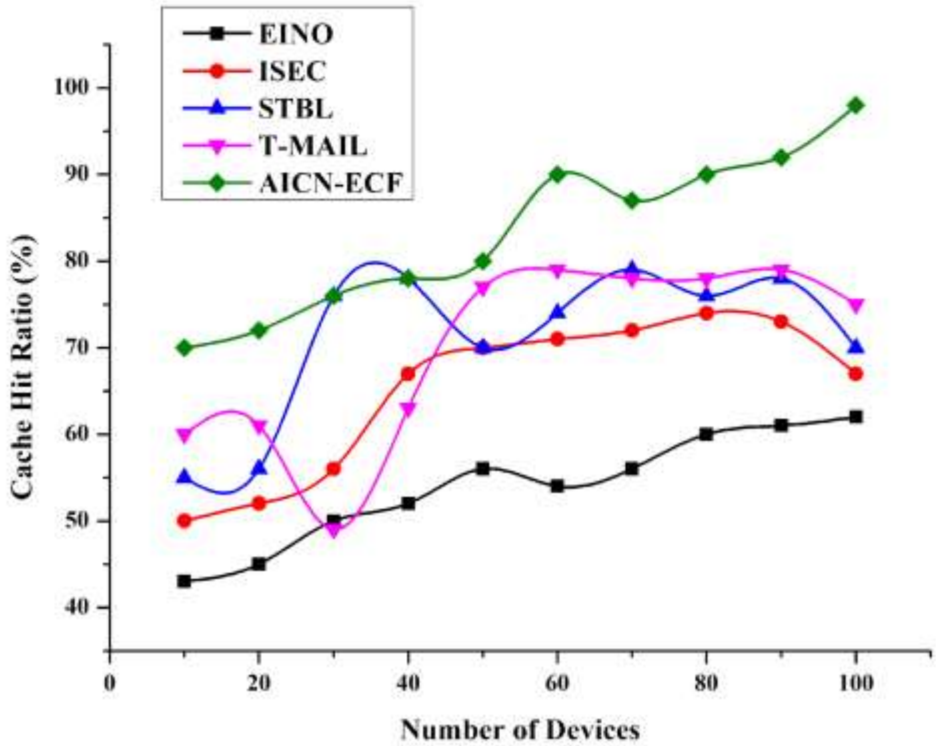

Figure 4: Cache Hit Ratio

(ii) Content Retrieval Time (CRT)

The proposed method improves performance by caching data close to the delivery channel, meaning additional end-user equipment in surrounding networks can get content quicker. The same pattern remains true for network cache routers; as cache sizes rise, the CRT decreases. Milliseconds is the unit of measurement for the period it takes to process content requests. Additionally, it is the amount of time that passes between submitting a content request and receiving a response. Certain variables, such as heavy network data traffic or a lack of varied content, may play a role. After assuming that diversity and congestion are almost equal, CRT may be used independently to evaluate the efficiency of caching techniques. Figure 5 shows the content retrieval time. It is computed utilizing the formula provided in equation (12).

$$CRT = interest\ travel\ delay\ +\ data\ travel\ delay \qquad (12)$$

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

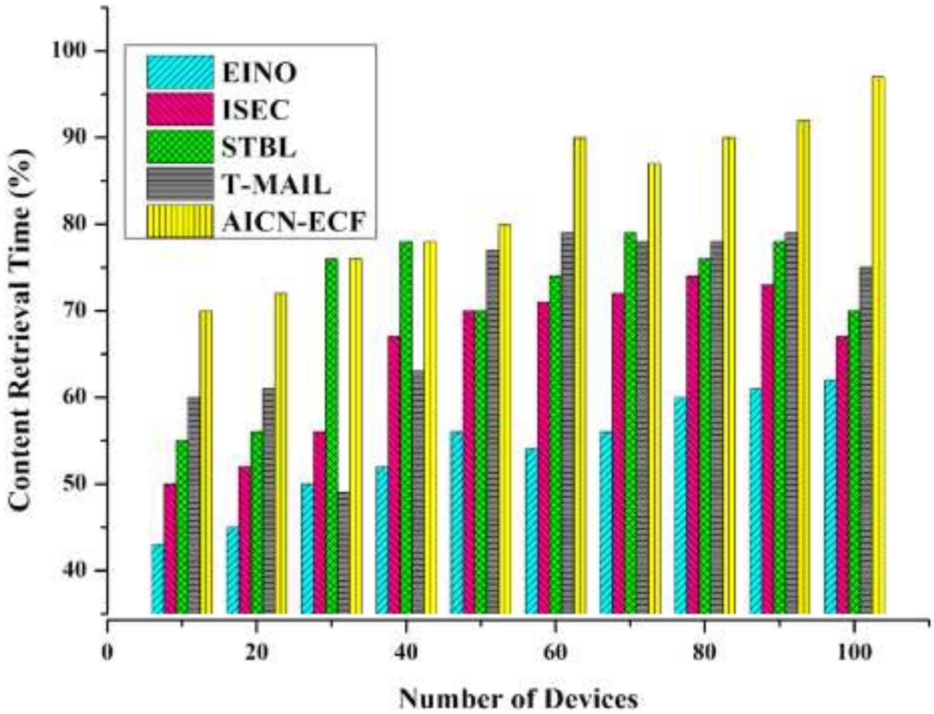

Figure 5: Content Retrieval Time

(iii) Data Security Ratio

Developing and deploying IoT devices requires high security due to the potential impact on the real environment. Security measures must be developed to permeate these systems to protect the nodes, networks, and data sent across them. While in-network caching improves network computing and caching capabilities, it also introduces new security risks, such as excluding production-level data from in-network access controls. The study's introduction of confidentiality-enhanced network coding—a combination of network encryption and the linear all-or-nothing transform led to the successful development of an access control scheme with efficient revocation support at edge networks. This study summarises and classifies the security and privacy issues for edge computing-based smart city applications. This research aimed to strengthen the security and privacy of smart city user data and reduce data traffic by expanding the environment for collaboration across edge servers. Smart healthcare applications in smart cities rely on ICN's increased data security to protect user privacy. Figure 6 shows the data security ratio.

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

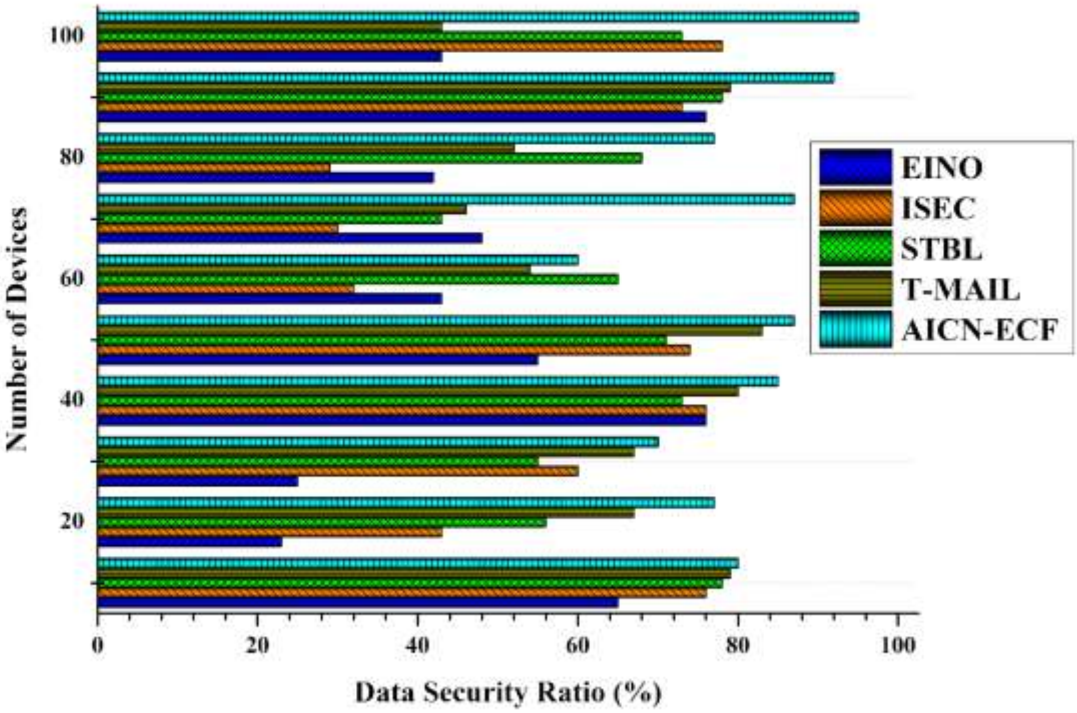

Figure 6: Data Security Ratio

(iv) Delay Ratio

A smart city's Internet of Things (IoT) system has increased energy consumption and network latency due to the existence of a cloud. Computing, storage, and networking resources are relocated to be closer to the data source in edge computing, built on a cloud computing architecture. Internet of Things (IoT) architecture is the same as cloud computing. Efficient energy consumption while preserving delay constraints is the key concern in edge computing while carrying out activities produced by IoT devices. This study provides an auctionable-based edge resource allocation mechanism to guarantee efficient resource computation for delay-sensitive tasks. To decrease content fetching latency with minimal hops visited, EC moves bandwidth, computation, and storage closer to the end users, limiting traffic at the network's core. Figure 7 signifies the delay ratio.

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

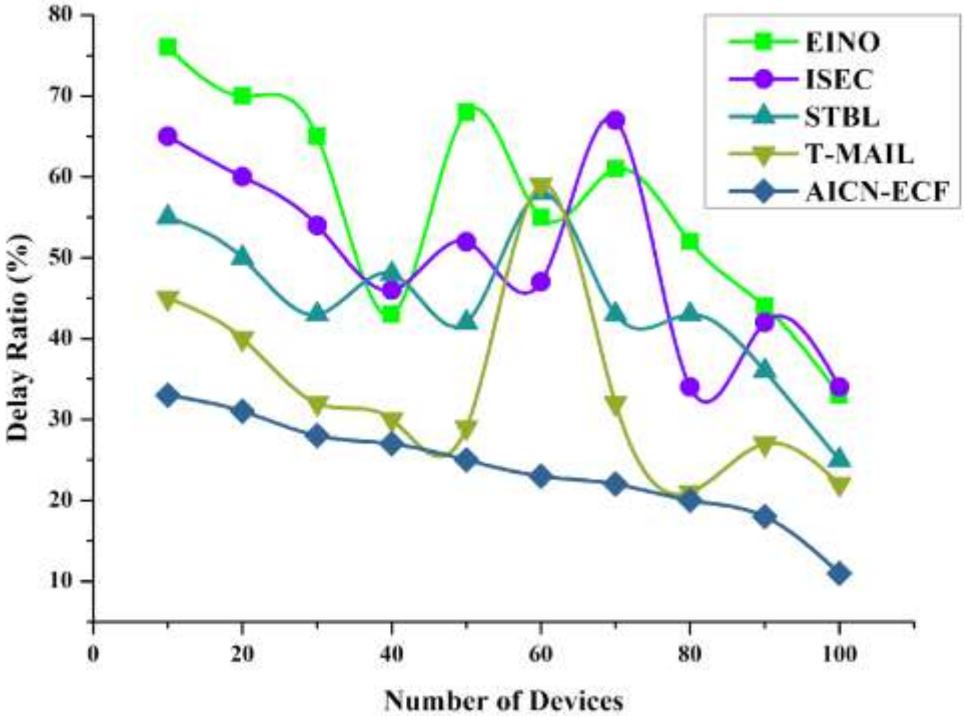

Figure 7: Delay Ratio

(v) Data Transmission Ratio

ICN can let EC nodes and cloud processing systems communicate sensory data quickly to support effective data analytics. Furthermore, by relying on the rapid resolution of named-based instances, ICN may enable the dynamic reconfiguration of smart city systems while reducing delays. Data availability in smart home apps may be improved, and redundant data transfers can be decreased using in-network caching. Consequently, ICN may provide direct content access, which lowers system overhead and improves data transmission efficiency. To further aid the mobility of users and Hospital-to-Home applications, ICN improves the user mobility functionality of networks. The unique benefits of ICN, such as more dependable and faster information transfer, make it an attractive option for this type of smart city application. Figure 8 shows the data transmission ratio.

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

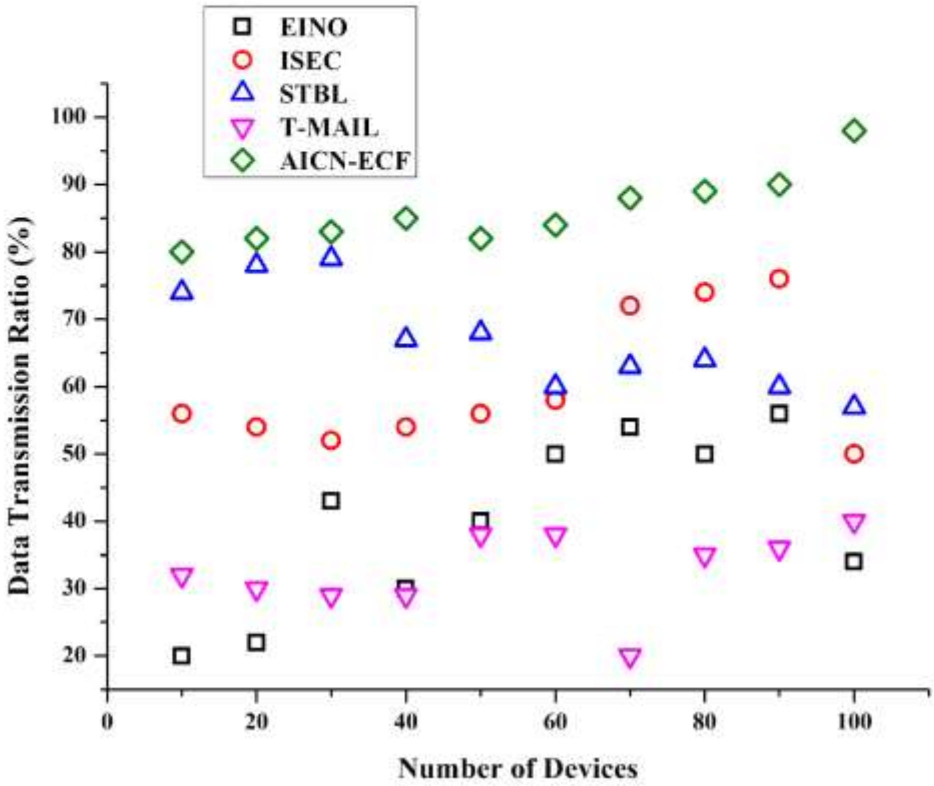

Figure 8: Data Transmission Ratio

5. Conclusion

This paper presents an Adaptive Information-Centric Network based on Edge Computing Framework (AICN-ECF) to reduce data traffic and latency with high security in smart cities. To tackle the problems with Smart City services, Edge Computing solutions need to examine the possibilities of moving some of the data processing from cloud data centres to edge devices and then dynamically rearranging the allocation of tasks based on the operating conditions, environmental factors, and purpose of the service. This article showed that purposefully built information and service models provide substantial benefits to achieve self-adaptive and composition-friendly Edge services. The main concerns with task allocation are finding the best spot for all tasks according to storage, processing power, and network bandwidth requirements and adapting to the ever-changing nature of the network. ICN over Edge computing is an effective method to guarantee a reduced response time. It will be useful in addressing security, mobility, naming, and massive data management issues. The information-centric character of IoT applications is an appropriate fit for ICN as well. Users of these applications would rather acquire content and services to connect more devices collectively. The experimental findings demonstrate that the suggested AICN-ECF model increases the cache hit ratio of 98.7%, content retrieval time of 97.8%, data security ratio of 96.5%, data transmission ratio of 95.6% and delay ratio of 11.2% compared to other popular models. However, edge servers' computing and transmission limitations can result in simplified services with reduced capability. Future studies will concentrate on edge computing task allocation, which will have to tackle the challenges of adapting to the unique characteristics of each network and finding a balance between competing objectives.

References

[1] Kuang, L., Gong, T., OuYang, S., Gao, H., & Deng, S. (2020). Offloading decision methods for multiple users with structured tasks in edge computing for smart cities. *Future Generation Computer Systems*, *105*, 717-729.

[2] Alsudani, M.Q., Jaber, M.M., Ali, M.H., Abd, S.K., Alkhayyat, A., Kareem, Z.H., and Mohhan, A.R., 2023. Smart logistics with IoT-based enterprise management system using global manufacturing. Journal of Combinatorial Optimization, 45(2).

[3] Ali, S.M., Elameer, A.S., and Jaber, M.M., 2021. IoT network security using autoencoder deep neural network and channel access algorithm. Journal of Intelligent Systems, 31(1), pp.95–103.

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

[4] Kumar, N., Vasilakos, A. V., & Rodrigues, J. J. (2017). A multi-tenant cloud-based DC nano grid for self-sustained smart buildings in smart cities. IEEE Communications Magazine, 55(3), 14-21.
[5] Chu, X., Jiang, H., Li, B., Wang, D., & Wang, W. (2020). Advances in Mobile, Edge and Cloud Computing. Mobile Networks and Applications, 1-3.
[6] Ali, M.H., Al-Azzawi, W.K., Jaber, M., Abd, S.K., Alkhayyat, A., and Rasool, Z.I., 2022. Improving coal mine safety with Internet of things (IoT) based Dynamic Sensor Information Control System. Physics and Chemistry of the Earth, 128.
[7] Xu, X., Liu, Q., Luo, Y., Peng, K., Zhang, X., Meng, S., & Qi, L. (2019). A computation offloading method over big data for IoT-enabled cloud-edge computing. Future Generation Computer Systems, 95, 522-533.
[8] Huang, J., Xing, C. C., Shin, S. Y., Hou, F., & , C. H. (2017). Optimizing M2M communications and quality of services in the IoT for sustainable smart cities. IEEE Transactions on Sustainable Computing, 3(1), 4-15.
[9] Ali, M.H., Jaber, M.M., Alfred Daniel, J., Vignesh, C.C., Meenakshisundaram, I., Kumar, B.S., and Punitha, P., 2023. Autonomous vehicles decision-making enhancement using self-determination theory and mixed-precision neural networks. Multimedia Tools and Applications.
[10] Sun, J., Sun, S., Li, K., Liao, D., Sangaiah, A. K., & Chang, V. (2018). Efficient algorithm for traffic engineering in Cloud-of-Things and edge computing. Computers & Electrical Engineering, 69, 610-627.
[11] Kadry, S., & Ghazal, B. (2019). Design and assessment of using smartphone application in the classroom to improve students' learning.
[12] Yin, Z., Chen, H., & Hu, F. (2019). An advanced decision model enabling two-way initiative offloading in edge computing. Future Generation Computer Systems, 90, 39-48.
[13] Anu Shilvya, J., George, S.T., Subathra, M.S.P., Manimegalai, P., Mohammed, M.A., Jaber, M.M., Kazemzadeh, A., and Al-Andoli, M.N., 2022. Home Based Monitoring for Smart Health-Care Systems: A Survey. *Wireless Communications and Mobile Computing*, 2022.
[14] Zhou, S., & Jadoon, W. (2020). The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment. Computer Networks, 107334.
[15] Zakaryia, S. A., Ahmed, S. A., & Hussein, M. K. (2020). Evolutionary offloading in an edge environment. *Egyptian Informatics Journal*.
[16] Ning, Z., Dong, P., Kong, X., & Xia, F. (2018). A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things. *IEEE Internet of Things Journal*, *6*(3), 4804-4814.
[17] Song, F., Xing, H., Luo, S., Zhan, D., Dai, P., & Qu, R. (2020). A Multiobjective Computation Offloading Algorithm for Mobile Edge Computing. *IEEE Internet of Things Journal*.
[18] Jošilo, S., & Dán, G. (2020). Computation Offloading Scheduling for Periodic Tasks in Mobile Edge Computing. *IEEE/ACM Transactions on Networking*, *28*(2), 667-680.
[19] Tong, Z., Deng, X., Ye, F., Basodi, S., Xiao, X., & Pan, Y. (2020). Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment. *Information Sciences*.
[20] Alelaiwi, A. (2019). An efficient method of computation offloading in an edge cloud platform. *Journal of Parallel and Distributed Computing*, *127*, 58-64.
[21] Zhang, F., Ge, J., Wong, C., Li, C., Chen, X., Zhang, S., ... & Chang, V. (2019). Online learning offloading framework for heterogeneous mobile edge computing system. *Journal of Parallel and Distributed Computing*, *128*, 167-183.
[22] Mu, S., & Zhong, Z. (2020). Computation offloading to edge cloud and dynamically resource-sharing collaborators in Internet of Things. *EURASIP Journal on Wireless Communications and Networking*, *2020*(1), 1-21.
[23] Chen, Z., & Wang, X. (2020). Decentralized computation offloading for multi-user mobile edge computing: a deep reinforcement learning approach. *EURASIP Journal on Wireless Communications and Networking*, *2020*(1), 1-21.
[24] Qiao, G., Leng, S., & Zhang, Y. (2019). Online learning and optimization for computation offloading in D2D edge computing and networks. *Mobile Networks and Applications*, 1-12.
[25] Li, C., Tang, J., & Luo, Y. (2019). Dynamic multi-user computation offloading for wireless powered mobile edge computing. *Journal of Network and Computer Applications*, *131*, 1-15.
[26] Alam, M. G. R., Hassan, M. M., Uddin, M. Z., Almogren, A., & Fortino, G. (2019). Autonomic computation offloading in mobile edge for IoT applications. *Future Generation Computer Systems*, *90*, 149-157.
[27] Yu, S., Langar, R., Fu, X., Wang, L., & Han, Z. (2018). Computation offloading with data caching enhancement for mobile edge computing. *IEEE Transactions on Vehicular Technology*, *67*(11), 11098-11112.
[28] Xing, J., Dai, H., & Yu, Z. (2018). A distributed multi-level model with dynamic replacement for the storage of smart edge computing. *Journal of Systems Architecture*, *83*, 1-11.
[29] Ismail, A. H., El-Bahnasawy, N. A., & Hamed, H. F. (2019). AGCM: active queue management-based green cloud model for mobile edge computing. *Wireless Personal Communications*, *105*(3), 765-785.
[30] Fan, W., Han, J., Chen, J., Liu, Y. A., & Wu, F. (2020). Probabilistic computation offloading and data caching assisted by mobile-edge-computing–enabled base stations. *Annals of Telecommunications*, 1-19.

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)

[31] http://iot.ee.surrey.ac.uk:8080/
[32] Arif, Z. H., & Cengiz, K. (2023). Severity Classification for COVID-19 Infections based on Lasso-Logistic Regression Model. International Journal of Mathematics, Statistics, and Computer Science, 1, 25–32. https://doi.org/10.59543/ijmscs.v1i.7715
[33] Dagistanli, H.A. (2023). An Integrated Fuzzy MCDM and Trend Analysis Approach for Financial Performance Evaluation of Energy Companies in Borsa Istanbul Sustainability Index. Journal of Soft Computing and Decision Analytics, 1(1), 39-49.https://doi.org/10.31181/jscda1120233

DOI: https://doi.org/10.54216/JISIoT.080208
Received: August 19, 2022 Accepted: February 24, 2023 (corrected version: January 17, 2024)