ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
Received: 7 September 2025; Accepted: 18 November 2025
https://doi.org/10.17559/TV-20250907002962
Original scientific paper

Reliable Resource Placement with Migration Function for Internet of Things (IoT) - Based Ubiquitous Wireless Network in Smart Cities

Shanmugham M.*, Chandrasekaran V.

Abstract: Smart City (SC) development with technological aspects depends on wireless communication and intelligent networks such as the Internet of Things (IoT). Wireless networks and IoT interconnect resources and projects them to be ubiquitous for various applications and user services. However, optimal placement of resources is mandatory to reduce their trivial usage and unavailability through migration features. This article proposes a Reliable Resource Placement (RRP) with Migration Function (MF) method to reduce the outage in SC communications. The outage due to overloaded network service demands is addressed with the help of a transfer learning network. The need for service migration and resource allocation intensity is decided by the learning network through continuous training. Based on the decisions, resource replacement for optimal service response is pursued through migrations. Therefore, the demand mitigation is eased through delayless migrations. This proposed RRP-MF reduces outage time by 13.79%, network overload by 14.04% and improves the response ratio by 13.41% for the maximum network load.

Keywords: IoT; resource migration; resource placement; smart city; transfer learning network

1 INTRODUCTION

Efficient placement of resources in Internet of Things (IoT) infrastructures for smart cities needs strategies that adjust to the changing needs of the environment, infrastructure, and services [1-5]. Decentralized IoT systems often struggle with balancing the computational demands of tasks that need quick responses. Resource optimization must consider various constraints such as energy limitations, distance to sensors, communication range, and the urgency of tasks [6]. Resources in a central location can cause traffic issues and reduce the ability to handle failures [7]. Flexible orchestration frameworks need to keep track of changes in data volumes and task density over time [8]. Using automated strategies with predictive modelling allows for proactive changes without the need for a central system to control everything [9, 10].

Network congestion in urban IoT communication is caused by things like frequent broadcasting, repeated data sending, and many sensors working at the same time during busy periods [11]. Static routing paths are not effective for handling the changing demands of smart services such as surveillance, transportation, and energy management [12, 13]. Additional load from protocols in publish-subscribe systems leads to unnecessary message repetition within clusters [14]. In multi-hop mesh networks, traffic that needs to be handled quickly has to compete with regular sensor data [15]. Differences in bandwidth between edge and cloud layers limit the ability of smart applications to communicate effectively [16]. Using predictive buffering and context-aware rebalancing at gateways is vital to reduce data loss [17].

Deep learning models offer strong support for IoT communication in smart cities by understanding the spatial and temporal relationships within distributed sensor networks [18]. Convolutional layers pick up on patterns of local interactions between nodes, while recurrent units deal with time-based traffic trends [19]. Graph-based neural networks show IoT structures with dynamic connections, allowing for better decision-making during periods of congestion [20]. Transfer methods help share insights between similar city areas, speeding up learning without requiring retraining [13, 21]. Multi-modal fusion combines inputs from audio, video, and environmental sensors to make better decisions in areas like traffic, security, and emergency services. Performance scalability ensures consistent performance across different devices by using adaptive model compression techniques [22, 23]. The contributions of the article are:

- The design and description of a novel resource placement method for ubiquitous wireless networks in smart cities
- The proposed method is experimentally analysed using hyperparameters and a simulation setup with external data inputs.
- The proposed method is comparatively analysed using outage time, average delay, network overload, response ratio, and resource utilization metrics.

2 BACKGROUND AND LITERATURE SURVEY

To help with smart scheduling for IoT services that have limits, Reddy et al. [24] made a mixed method that uses particle swarm optimization along with deep Q-learning. The method makes processing more accurate, cuts down on delays, and helps spread out the workload better. Reddy et al. [25] came up with EFLsm, a layered method that uses swarm intelligence for managing resources well. The results show that less energy is used and services run better in smart places.

To make sharing of edge resources fairer, Sahoo et al. [26] made a learning automata approach that listens to what users need and how resources change. The method lowers the average time for responses and helps the method grow smoothly. Wan et al. [27] designed a dynamic resource manager that uses deep reinforcement learning and mobile edge computing. The model makes things more efficient for tasks that need fast responses in smart cities.

To handle big data syncing, Adreani et al. [28] made a digital twin framework that connects different data sources in real time for smart cities. The results show it is easy to set up and help people interact more with the city in different ways. Jeevanantham et al. [29] made a neuro-fuzzy routing model that helps choose the best paths for sensor communication. Testing shows better packet delivery and longer life for the network devices.

To make the power grid respond quickly, Singh et al. [30] built a smart energy method that uses deep learning and IoT devices. Tests show lower chances of power cuts and better grid performance. To work together in spread-out networks, Gou et al. [31] developed a cooperative method with several fog nodes. Results show faster responses and less use of resources for smart services.

Bangash et al. [32] created a model for placing software controllers in a virtual environment. The model improves consistency and reduces the extra work for control messages. To make smart city computing better, Sun et al. [33] developed a resource allocation method that uses attention for edge computing. Test results show better performance, stability, and shorter delays.

Cities et al. [34] developed an AI technique that matches city resources with sustainability goals. The method uses machine learning to guess how much demand there will be and adjusts services like waste, lighting, and traffic. Results show less pollution and more cost savings. To improve city supply chains, Kapanski et al. [35] used geospatial clustering to map and improve systems like water and power. This clustering allows for quicker action and better planning for the future.

Papaioannou et al. [36] introduced a framework that organizes edge and cloud nodes in a hierarchy. The model uses less extra work while handling different device types. To make service delivery easier, Singh et al. [37] made a model that brings together fog and cloud parts for access to city resources. A scheduler handles priority changes for tasks across different levels.

Peshattiwar et al. [38] developed a method that moves IoT microservices between the edge and cloud. The results show better service availability and improved user experience. Mohamed et al. [39] built an IoT platform to manage things like roads, streetlights, and sewers. The method shows cost savings and better efficiency in city operations.

Bellini et al. [40] made a shared model for smart city platforms that uses different knowledge systems. The model makes things more consistent across different areas.

Traditional resource management methods often fail to adapt to complex network conditions and experience sudden failures. It may be due to inefficient resource utilization and delayed service delivery to end users with excessive overloads. A Reliable Resource Placement with Migration Function (RRP-MF) is proposed to ensure optimal resource placement and service migration. It improves load balancing between nodes and end-user experience even under varying network conditions.

3 PROPOSED WORK

The description of reliable resource placement is provided in this section with the process flow illustration. The network incorporation, resource placement, and migration processes are described in the flow. The view of the proposed method is given in Fig. 1.

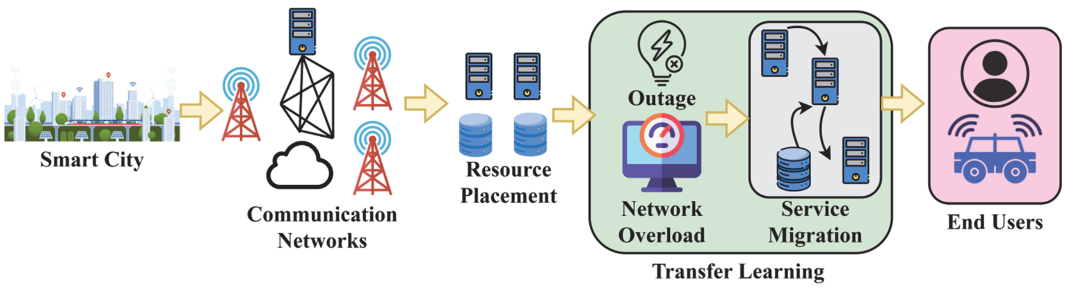


Figure 1 View of the proposed RRP-MF

The proposed method incorporates the heterogeneous wireless network to facilitate communication between users. These networks experience overloads and, therefore, outages due to varying users. To handle these problems, optimal resource placement and service migration functions are introduced. These functions are validated using a transfer network for learning different network situations. Based on the learning process, the decisions on service migrations and resource placement are made. The resource placement in the communication network targets the mitigation of outage and overloads in the network. Based on the overloads, the service migrations are performed. Such migration ensures maximum end user responses with confined outages. This provides a solution to ensure maximum resource availability through migration. The overload and outage assessment with migration recommendations are defined using transfer learning (Fig. 1).

The process of resource placement in smart cities improves the performance of the communication system to ensure reliable service delivery. The proposed RRP-MF analyses the communication structure and topology to identify connected nodes. The nodes are checked for availability to place sources based on priority. Multiple resources are placed in high-traffic zones, and they are interconnected to balance the load. The connection of the communication network in the smart city is expressed as C_{comm} in the below equation.

$$C_{\text{comm}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \left[\frac{\left(d_{ij} \times \beta_{ij}\right)}{L_{ij} + e} \right] \times \gamma_{ij} \tag{1}$$

The communication network's efficiency is evaluated by analyzing its interaction with various nodes and devices. The number of nodes in the network is denoted as (n) and devices connected within the network are represented as (m) The data demand experienced by the network at the node (i) and device (j) is defined as d_{ij}. The bandwidth available for a network is denoted as β_{ij} which helps to predict the potential data range within the network. The latency experienced by the network during communication is represented as L_{ij}. An error occurred during network verification is captured as (e) and is used to prevent external loss. The dependency of a specific network path during communication is denoted as reliability and is represented as (γ_{ij}). This helps to identify optimal network paths suitable for communication. It comprises user demand as $[d(\mu)]$ and computational complexity (px) with network load (nd).

$$\begin{aligned} \lambda[d(\mu), px] &= -\beta_{ij} \times d(\mu) - [O(C_{\text{comm}} \times \beta_{ij})] - d_{ij} \\ &= -d(rs) \times (d(\mu) - e) - [O - (C_{\text{comm}} - L_{ij}) - d_{ij}] \\ &= d(rs) \times (e - \gamma_{ij}) - d(\mu) \rightarrow [d_{ij} > L_{ij}] \end{aligned} \tag{2}$$

$$\begin{aligned} \lambda[d(\mu), nd] &= [d(rs) + (\beta_{ij} \times \gamma_{ij})] \times (d(\mu) - C_{\text{comm}}) \times nd \\ &= \max(px) - \beta_{ij} = \left(\frac{d(rs)}{d(\mu)}\right) - \max(nd) - 1 \\ &= d(rs) \times [px + nd] - \beta_{ij} \times [O(C_{\text{comm}}) + d(\mu)] \end{aligned} \tag{3}$$

The resource placement decision to select the optimal node is calculated as (λ) in which $\lambda[d(\mu), px]$ in Eq. (2), performs resource placement. The derivation of $\lambda[d(\mu), nd]$ in Eq. (3) describes optimal node selection for resource placement. In Eq. (3), the demand for resource $\lambda[d(\mu), nd]$ is denoted as $d(rs)$ which is calculated by processing the (β_{ij}), (γ_{ij}) and (C_{comm}). If the (d_{ij}) is greater than the $d(rs)$, the system initiates a request to select needs. This helps to identify whether the selected node is optimal for resource placement or not. The node selection is evaluated by incorporating (nd) for (px) in $\lambda[d(\mu), nd]$. It helps the network to select a node based on varying $d(rs)$ and (nd) that leads to overload. It calculates which node in the network provides the maximum and minimum (C_{comm}) that causes communication overload as $[O(C_{\text{comm}})]$ during resource allocation. This condition for node and resource selection for placement is computed as $F(\lambda)$ in the following equation.

$$F(\lambda) = \begin{cases} \text{if} \quad d(\mu) > d(rs) \rightarrow O(C_{\text{comm}}) \le \min C_{\text{comm}} \\ \text{if} \quad \lambda[d(\mu), nd] \rightarrow \left(\frac{d(rs)}{d(\mu)}\right) = 1, O(C_{\text{comm}}) = \max C_{\text{comm}} \end{cases} \tag{4}$$

In smart cities, the network selects the idle node for placement if $d(\mu) > d(rs)$ with maximum (γ_{ij}). It also maximizes the (β_{ij}) to minimum $[C_{comm}]$ to align requested $d(\mu)$ to the network. This reduces (px) arises during resource placement and selects a suitable resource for the network. The misplacement of resources is reduced by verifying $\lambda[d(\mu), nd]$ with $\left(\frac{d(rs)}{d(\mu)}\right) = 1$ ensures that the node is suitable for placement.

If $\lambda[d(\mu), nd] \rightarrow \left(\frac{d(rs)}{d(\mu)}\right) = 0$ indicates that the node is not suitable for resource placement. It avoids computational complexity due to excessive (nd) to improve allocation capacity. If the $d(\mu)$ is high, then the selected node for resource placement is suitable and the $d(\mu)$ is less than the node and is free and reliable for future placement. In the presentation below, the resource placement process is illustrated.

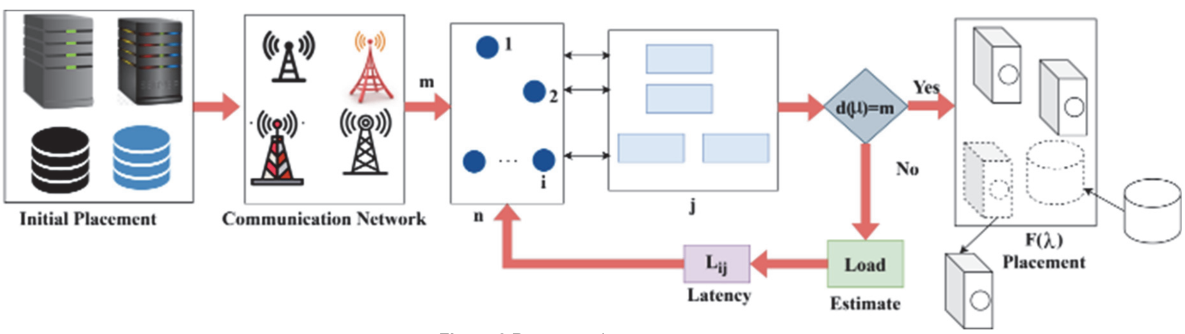


Figure 2 Resource placement process

The illustration of $d(\mu)$ placement is depicted in Fig. 2. The m is the key factor for connecting n and j for handling user demands. A balance between the same is confirmed using (nd) and γ_{ij} for maximum response distribution. In the m based allocation, the $[d(\mu) > m]$ is the $F(\lambda)$ placement condition for retaining C_{comm} such that L_{ij} is the actual delay identified. This connectivity between i and j is used to ensure high m to reduce $d(\mu)$.

Outage and Overload Detection

Once the resources are placed based on $F(\lambda)$ from $\lambda[d(\mu), nd]$ and $\lambda[d(\mu), px]$, the network processes the detection of outage and overload. It incorporates node availability, response time, processing delay, and traffic levels to analyze the outage conditions. If the node fails to respond within a particular time or its capacity is zero can lead to an outage. Overload in the network is detected by comparing the current load (nd) to its maximum $[C_{\text{comm}}]$. The outage in the network is evaluated as (Dg) to maintain optimal communication for end users.

$$Dg = \sum_{i=1}^{n} \left[\begin{array}{l} \left(1 - \frac{h_i(t)}{\min(t)}\right) + \left(\frac{L_i}{\max(L)}\right) \\ + \left(1 - \frac{\gamma_i}{\max(\gamma) - \min(\gamma)}\right) \end{array} \right] \tag{5}$$

The nodes available in the network are denoted as (h) and the minimum available node is considered as min (h) at a time (t). If $h_i(t) < \min(L)$ then, the system defines the node as inactive and $h_i(t) > \min(h)$ then the node is accessible. The latency may increase the delays concerning $[\max(L)]$ to indicate failures. The actual reliability and average reliability are monitored from the difference of $[\max(\gamma) - \min(\gamma)]$ to minimize service delays. If this (Dg) exceeds its particular limit based on (px), then the node is considered for migration or rerouting. The network overload (nod) is derived by monitoring actual $[C_{\text{comm}}]$ and (nd) in the following equation.

$$nod = \sum_{i=1}^{n} \left[\left(\frac{nd_i(t)}{\beta_i(t)}\right) + \left(\frac{q_i(t)}{\max(q)}\right) + \left(1 - \frac{cp_i(t)}{\max(cp)}\right) \right] \tag{6}$$

The load is applied to the node to increase the actual (nd) based on $\beta(t)$ is identified to prevent the impact of service reliability. The queue length of the node that leads to buffering is denoted as (q_i) and its maximum value is monitored as $[\max(q)]$. The high (q_i) increases the latency and network congestion in the network. The capacity of a node is represented as $[cp(t)]$ and maximum capacity handled by (i) is denoted as $[\max(cp)]$. The calculation of $\left(1 - \frac{cp_i(t)}{\max(cp)}\right)$ ensures the capacity of the node and its overall utilization. A high (nod) node should be offloaded to maintain optimal performance in the RRP-MF method. The total communication performed within the time interval (t) is defined based on input (nd) sample and sequence of (m) as $\left[\hat{C}_{\text{comm}}(t)\right]$ in the below.

$$\begin{array}{l} nd = [\lambda(d(\mu), px) - \lambda(d(\mu), nd)], \text{ load} \\ \iota = \left[\left(\frac{Dg}{nod}\right) \times cp\right] \forall d(rs), \text{ communication} \\ \hat{C}_{\text{comm}}(t) = (nd \times m) + (k + O(C_{\text{comm}})) \end{array} \tag{7}$$

The communication observed at the initial load of the node is monitored in (nd). The involvement of an interconnected device for communication is analyzed based on $d(rs)$ to reduce the delay in (C_{comm}) during high demand conditions. The overall communication at a particular interval helps to understand the incoming requests. It is useful in managing (Dg) and (nod) to perform further migration based on need. In a smart city network, the (Dg) and (nod) is to be normalized before the process of migration to ensure smooth communication. It is computed as $(S_n(t))$ to balance the influence of factors that affect the migration.

$$\begin{array}{l} S_{i1}(t) = C_{\text{comm}}\left(F(\lambda) \hat{+} O(C_{\text{comm}})\right)_1 \\ S_{i2}(t) = 2\left(\hat{C}_{\text{comm}}(t) - [F(\lambda) + O(C_{\text{comm}})]_2\right)\left(\frac{Dg}{nod}\right) \forall h. \\ S_n(t) = \hat{C}_{\text{comm}}(t-1) - [F(\lambda) + O(C_{\text{comm}})]_t - \left(\frac{Dg}{nod}\right) \forall h(t-m) \end{array} \tag{8}$$

$$\begin{array}{l} \text{if} \quad (S_n(t) \geq m_{\text{stable}}) \forall (nd = m) \\ \text{if} \quad (S_n(t) < m_{\text{stable}}) \forall (nd \neq m) \end{array} \tag{9}$$

The combines $\left[\hat{C}_{\text{comm}}(t)\right]$ with $F(\lambda)$ helps the system to decide the migration condition based on the normalization of (nod) and (Dg). It evaluates each device (m) based on their node availability (h) within the time interval (t) and the number of nodes $[i = 1, 2, ..,]$. It decides the normalization if the actual $\left[\hat{C}_{\text{comm}}(t)\right]$ is sufficient even under high (nod) and (Dg) that delays the migration. In Eq. (9), the system determines the need for migration based on the condition. If $\left[S_n(t) \geq m_{\text{stable}}\right]$ and the network load (nd) is equal to the total number of devices with a stable device constraint (m_{stable}). If $S_n(t) < m_{\text{stable}}$ and $(nd \neq m)$ then the system initiates migration to prevent service disruption. The relation between the demand and overload for service migration is depicted in Fig. 3.

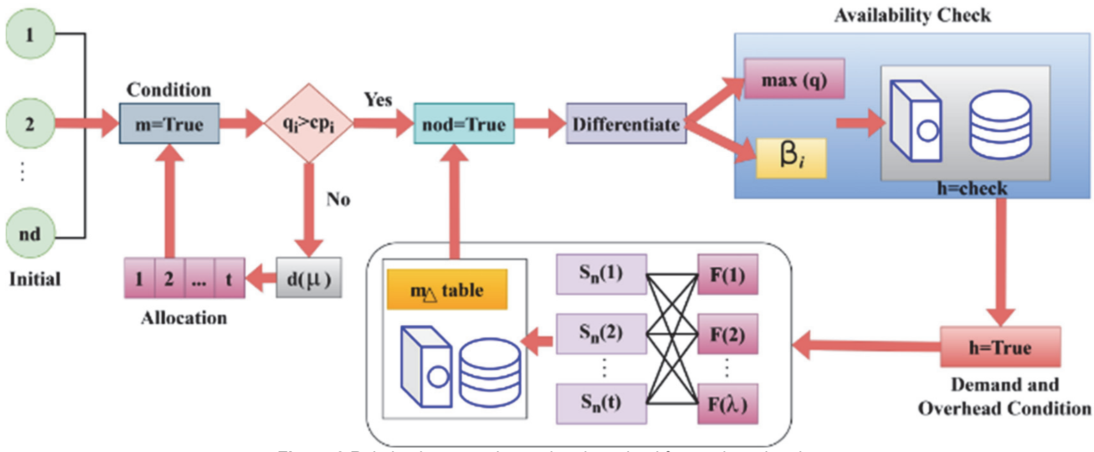


Figure 3 Relation between demand and overload for service migration

The relation between *nd* and *nod* is illustrated in Fig. 3. This relation between the factors is a necessary validation to maximize resource allocation and migration concurrently. Based on the available *t*, the q_i exceeding the cp_i are separated to distinguish the actual β_i from $\max(q_i) \in t$. This differentiation increases the verification need of h such that its true (Available) maps the $S_n(t)$ and $F(\lambda)$. The placement before and after migration is eventually used to ensure the *nod* to interfere with the network for its actual load. Therefore, these t identifications for resource allocation/placement and migration are mandatory using the m_{stable} output in any $nod = true$ condition. To maintain uninterrupted service delivery in the smart city network, service migration is important and is discussed in the following.

Service Migration for High Response

The migration distributes nodes based on their performance and reallocates service to improve response to end users. The transfer network monitor services across all nodes and detects (Dg) and (nod). The data is transferred to the target (n) for routing to ensure minimal disruption with high response. If the node is unstable, then the service is paused to balance the optimal (rs) utilization. It equalizes these inputs to regulate the effect of these inputs on the migration choice and employs continuous training to adjust to the changing network conditions. Transfer learning model assesses the difference between the present and optimum load parameters in order to determine quantitatively the necessity of migration. The transfer network will guarantee a balance in the load distribution by prioritizing areas with high demand of services and evaluating temporary migrations. Its approach to training is based on the update on a real-time network data, which allows it to forecast overloads and outages, so it becomes possible to cause a migration only in necessary cases. This learning system is able to maximize response ratios, reduce delays, and ensure dependable communication of smart city IoT networks. The below equation $\left[Zm(t)\right]$ computes the migration process,

$$\gamma_{\text{mig}} = \left[\frac{(nd)-(px)}{(nod)} + \left(\frac{Dg}{S_n(t)}\right)\right] \forall \left(S_n(t) < m_{\text{stable}}\right) \tag{10}$$

$$Zm(t) = \left[\left(\frac{\hat{C}_{\text{comm}}(t) + F(\lambda)}{(nod)+(Dg)}\right) \times \left(\frac{d(\mu)-d(rs)}{O(C_{\text{comm}})}\right) \forall \left[\gamma_{\text{mig}} = 1\right]\right] \tag{11}$$

Eq. (10) decides the need for migration and triggers migration only if $\left[S_n(t) < m_{\text{stable}}\right]$ with $\left[\gamma_{\text{mig}} = 1\right]$. It evaluates the difference between (nd) and (px) with (nod) to allocate resources to each node. A high value of γ_{mig} indicates immediate migration and a low value avoid migration. The actual migration is performed in Eq. (11) and triggers $Zm(t)$ if $\left[\gamma_{\text{mig}} = 1\right]$ to transfer the service data. A high $Zm(t)$ indicates that the system is in a complex situation, and immediate migration is performed to maintain a reliable smart city experience. The constraints for migration are detailed in the following equation.

$$Zm(rs) = \left[\frac{(cp-q)}{\beta(t)} + \hat{C}_{\text{comm}} \times \left(\frac{px}{Dg}\right)\right] \forall \left(\gamma_{\text{mig}} = 1\right) \tag{12}$$

$$Zm(nd) = \left[\frac{(nd-nod)}{d(rs)} + \left(\frac{nd \times m}{F(\lambda)+d(\mu)}\right) \times \left[\begin{matrix} O(C_{\text{comm}}) + \\ S_n(t) \end{matrix}\right]\right] \tag{13}$$

The migration performer to maintain high (rs) regions is computed as $Zn(rs)$ in the Eq. (12). It ensures that the migration is directed to regions that are in urgent service need. This prioritizes users based on $d(\mu)$ during resource placement. The temporary migration for (nd) is evaluated as $Zm(nd)$ in Eq. (13). The migration process is depicted in Fig. 4.

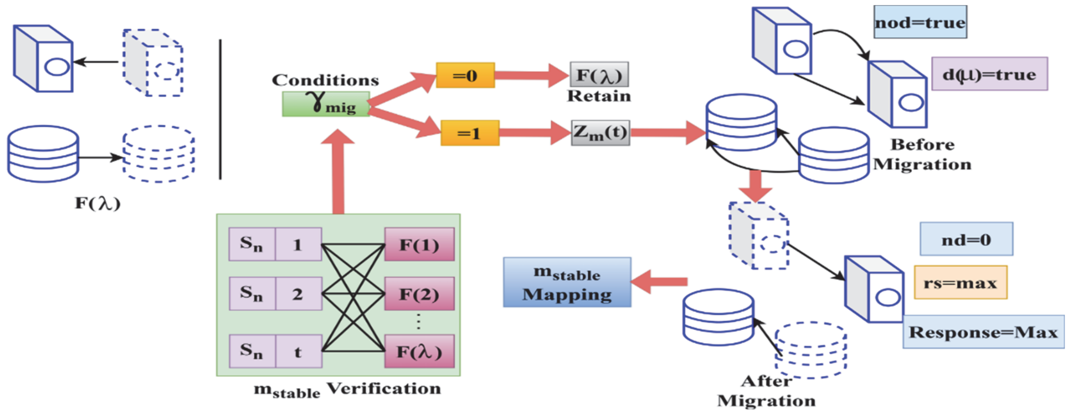


Figure 4 Migration process depiction

The decisions on migration rely on $F(\lambda)$ and $Z_m(t)$ based on multiple t identified. The $S_n(t)$ and $F(\lambda)$ is the initial combination for maximizing optimality through nod and nd management. The $Z_m(t)$ is proceeded with the following targets of $nod = \text{true}$ and $d(\mu) = \text{true}$ that are being experienced. Therefore, to reduce the migration constraints, the $\gamma_{mig} = 1$ is the validated condition. This requires the $F(\lambda)$ and β_i for maximum responses to stabilize $\gamma_{\text{mig}} = 1$. This process requires $nd = 0$ and response as maximum through $d(\mu)$ mitigation. Therefore the $(q_i > cp_i)$ condition check is reduced to ensure maximum $F(\lambda)$ such that migration eases stability.

The quantitative assessment of the migration decision is by establishing the value difference between the present and optimum load parameters such that a large value suggests that there has to be migration. Real migration implies transfer of service data to target nodes and priorities given to regions with high service demands and temporary migration assessed to stabilize the network. The transfer learning network will continually observe several variables to make sure that migration reduces overload, optimizes response ratios and reduces delays. This responsive and dynamic strategy reallocates the services of overloaded or broken nodes to healthy ones ensuring efficient and stable communication of smart city networks.

(Fig. 4). The reliable placement of nodes improves (rs) to end users is computed as $Um(rs)$ in the equation.

$$Um(rs) = \min\left[\frac{L(t) \times Dg}{nod}\right] \times \left[\frac{zm(rs) + zm(nd)}{\hat{C}_{\text{comm}}(t) \times S_n(t)}\right] + (d(\mu) - d(rs)) \times (1 - C_{\text{comm}}) \tag{14}$$

The overall delay in communication is minimized by reducing $(L(t))$ and (Dg) based on the $d(rs)$. It selects the optimal path by considering available (rs) with minimal (nod). This helps the system manage high-end user response to their demand and requests with high resource allocation under optimized decision-making for smart city communication. The proposed RRP process is described in Algorithm 1.

Algorithm 1 RRP Process

Input:

Network state data: nd, h, $\max(cp)$, n

Output:

- Resource placement decisions
- Service migration actions

Begin

Initialize transfer learning network TLN

Continuously monitor network parameters:

For each node i in network:

$$nod = \sum_{i=1}^{n}\left[\left(\frac{nd_i(t)}{\beta_i(t)}\right) + \left(\frac{q_i(t)}{\max(q)}\right) + \left(1 - \frac{cp_i(t)}{\max(cp)}\right)\right]\left(1 - \frac{cp_i(t)}{\max(cp)}\right)$$

$$S_n(t) = \hat{C}_{\text{comm}}(t-1) - \left[F(\lambda) + O(C_{\text{comm}})\right]_t - \left(\frac{Dg}{nod}\right) \forall h(t-m)$$

For each node i:

$$\gamma_{\text{mig}} = \left[\frac{(nd) - (px)}{(nod)} + \left(\frac{Dg}{S_n(t)}\right)\right] \forall \left(S_n(t) < m_{\text{stable}}\right)$$

$$Zm(t) = \left[\left(\frac{\hat{C}_{\text{comm}}(t) + F(\lambda)}{(nod) + (Dg)}\right) \times \left(\frac{d(\mu) - d(rs)}{O(C_{\text{comm}})}\right) \forall \left[\gamma_{\text{mig}} = 1\right]\right]$$

Evaluate migration necessity:

For each node i:

If $\left[S_n(t) < m_{\text{stable}}\right]$ Then

$Zm(rs)$ =True

Else

$Zm(nd)$ =True

For each node i :

$$Um(rs) = \min\left[\frac{L(t)\times Dg}{nod}\right]\times\left[\frac{zm(rs)+zm(nd)}{C_{\text{comm}}(\hat{t})\times S_n(t)}\right]+$$

$$\left(d(\mu)-d(rs)\right)\times\left(1-C_{\text{comm}}\right)$$

If $Zm(rs)$ =True Then

$$h=1$$

$$\left[\gamma_{\text{mig}}=1\right]$$

$$(q_i > cp_i)$$

Perform migration:

- Transfer service data from node i to node j

$$Um(rs) = \min\left[\frac{L(t)\times Dg}{nod}\right]\times\left[\frac{zm(rs)+zm(nd)}{C_{\text{comm}}(\hat{t})\times S_n(t)}\right]+$$

$$\left(d(\mu)-d(rs)\right)\times\left(1-C_{\text{comm}}\right)$$

Check if temporary migration improves load distribution
If yes, proceed; else revert migration
Update network state and TLN model continuously
End

4 RESULTS AND DISCUSSION

Experimental Setup and Data Description

The proposed RRP is analyzed using NetSim experiments considering a 1200 × 400 m^2 smart city region. The network with mesh topology is constructed using the infrastructure devices disclosed as follows. The region is installed with a central cloud server accessible through 11 infrastructure units to support 300 users. The conventional telemetry transport and constraint application protocols are HTTPS used for communicating requests and responses through packet data. The users communicate with the infrastructures using 802.11 and 802.15.4 standards for request and response sharing. The network experiences a load between 50 and 600 requests/ interval for service. The minimum outage time for disconnection is 1.2 seconds, after which the communication interval is to be re-initiated [41]. The bandwidth utilization varies from 1 Mbps to 500 Mbps based on the device used. The data utilized from the referenced source contains temporal smart city data related to e-Health and smart industry. Defining the human-activities in these industries, request and response traffic with fluctuations dropouts and timed resources are temporaly collected. The data is accumulated at an interval of 60 seconds for 5 days based on the response and request packet count. Based on this data, the temporary network disconnection and service responses are computed for a continuous period of 10 intervals.

Metrics Discussion

The comparison results for outage time, average delay, network overload, response ratio, and resource utilization metrics are presented with a discussion in this section. The results are obtained by varying the number of users (20 to 300) and the network load (requests/ interval) (50 to 600) parameters. The proposed RRP-MF is compared with ERA-LA [26], RoISM [24], EFLSM [25], and ESGCA [31] methods described in the related work section.

Outage Time

In the proposed RPM-TN method, the outage time is reduced by normalizing overload and outage conditions using communication interval tracking. This method redirects services through stable paths and available nodes to prevent overloading only if the network load increases.

It helps the system to redistribute load during critical intervals and minimize outage time even under high user demands (Fig. 5).

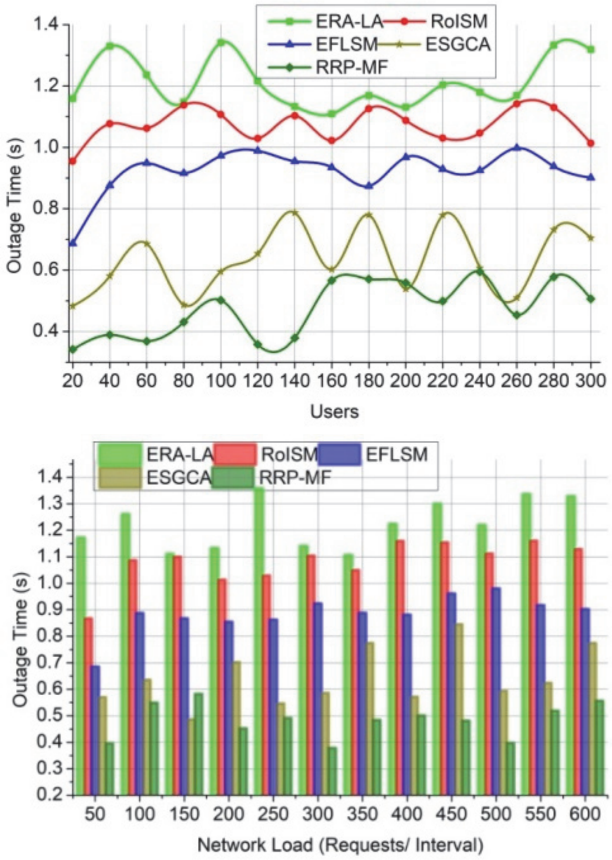
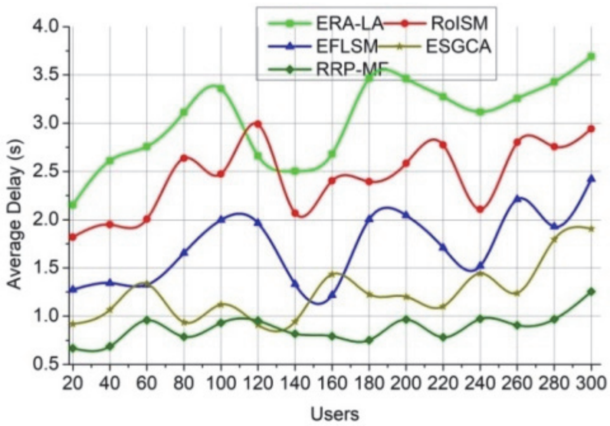


Figure 5 Outage time

Average Delay

In smart cities, the interruption of latency and communication affects the network performance. The system routes the request based on high-response and low-latency with the help of transfer learning. This helps the system to pre-plan its resource placement within the smart city to ensure minimal average delay across varying conditions (Fig. 6).



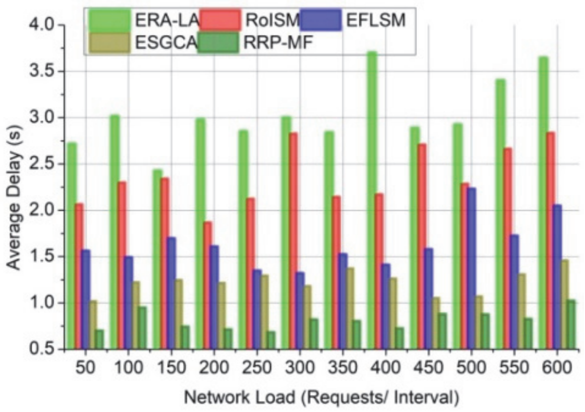


Figure 6 Average delay

Network Overload

Network overload measures the number of requests exceeding the network's capacity to process within a given interval. The proposed method minimizes overload by performing early detection through overload estimation and optimal migration.

It allows the RPM-TN method to balance the traffic across nodes and minimize overload spikes within the network. This reduces network overload to maintain smooth communication in smart cities (Fig. 7).

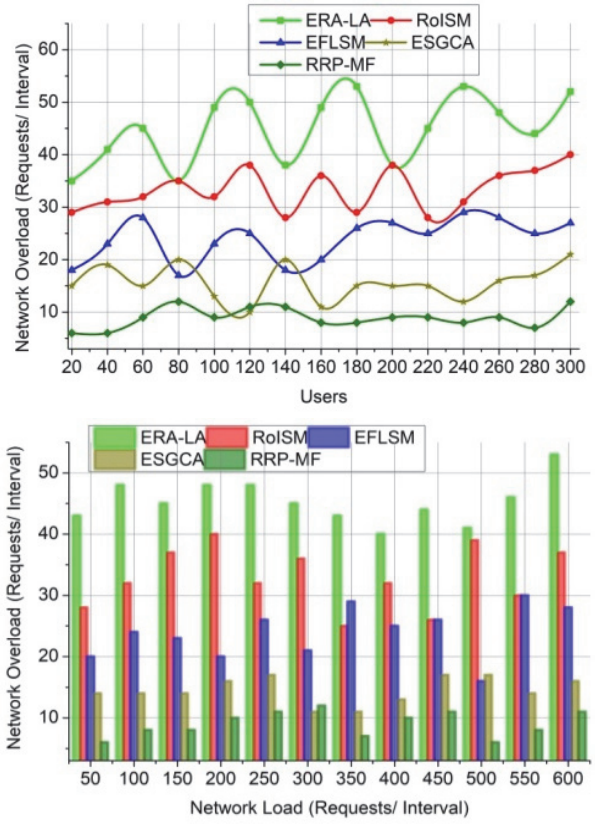


Figure 7 Network overload

Response Ratio

A high response ratio indicates a low service drop rate with high system reliability. The transfer network monitors services and detects outages and overload to ensure minimal disruption. Migration to high-response zones with minimum latency enhances the overall communication and leads to achieving a higher response ratio across all network conditions (Fig. 8).

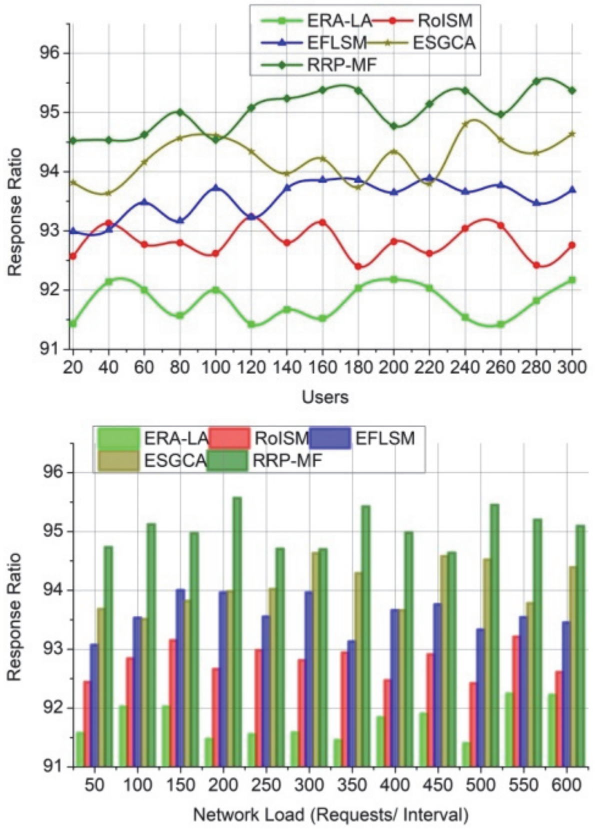


Figure 8 Response ratio

Resource Utilization

The high resource utilization reduces idle time and maximizes operational efficiency during fluctuating loads. Optimal migration decisions maintain a balance between user demand and requests for optimal resource utilization. This high resource utilization improves the efficiency and reliability of the communication network (Fig. 9).

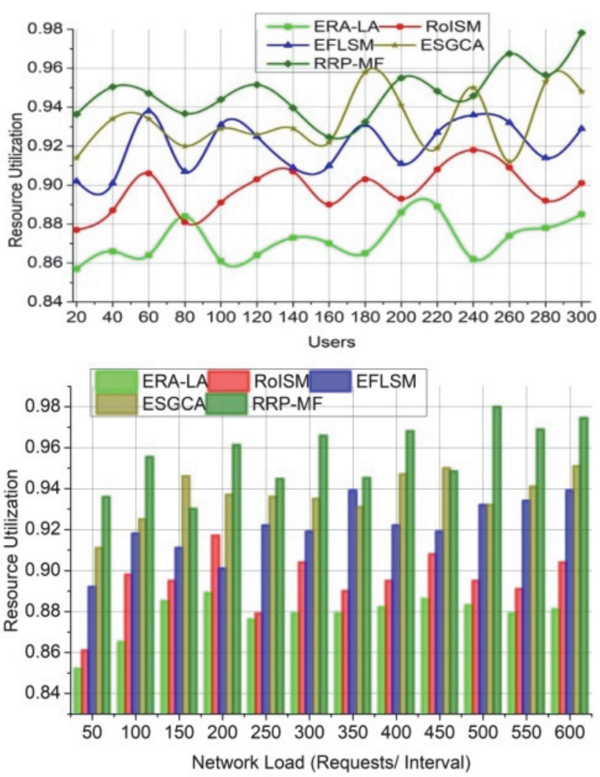


Figure 9 Resource utilization

Computation Complexity

The RRP-MF proposed is less complicated since it uses a transfer learning network to regularly measure and balance various parameters of the network, including node load, availability, latency, and traffic conditions. This normalization balances the effects of the different factors preventing unnecessary migrations and computation overload. The computation complexity analysis is presented in Fig. 10.

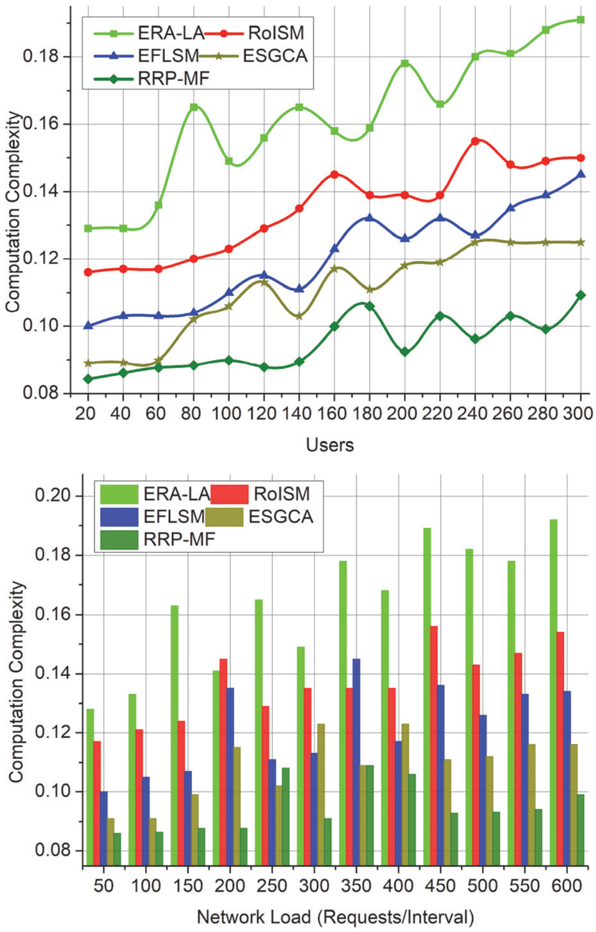


Figure 10 Computation complexity

The approach applies quantitative assessment of the load variation in order to initiate migration decisions when necessary without unnecessary resource redistribution. Selective assessment of temporary migrations is done to ensure that load balance is maintained without bringing extra migration activities. The system is simplifying the processes of decision-making and optimizing the distribution of resources not only because it has prioritized urgent needs of the services and updated dynamically with continuous training with up-to-date data, but also because it has minimized the complexity of computations, which guarantees the optimal performance of the network and its reliability.

The proposed approach helps to minimize outage by constantly assessing network resources like node availability, response time, processing delay, and traffic load. Normalization of the overload and outage parameters enables the system to balance the evaluation to ensure that the system redirects the services in stable routes and available nodes before the service disruption effects take place. This proactive migration minimizes the time of outage to avoid a case where the overloaded or failing nodes can trigger a prolonged communication breakdown. Response is enhanced by giving preference to migration of services to regions with high priorities in service requirements and service transfer to stable nodes that have low latency and enough bandwidth. The transfer learning is constantly informed of the network data and therefore, it predicts overloads and outages to cause timely migrations. Such a dynamic reallocation of services maximizes the response ratio as the services are more available and the delays are minimized to the end users in the smart city networks. The load condition of nodes is early detected and estimated to allow the optimum migration decision that will redistribute the load to maintain a balance between nodes. The system checks the disparity of the load parameters at present and maximum load and initiates migration in the case of surpassing the limits. Temporary migrations are also evaluated to keep a balance in the networks without incurring unnecessary overheads. This will cut the number of requests surpassing network capacity hence overload is reduced and smooth communication is ensured amidst changing user demands and network loads. The results of the metrics comparisons are tabulated below. The final value of the comparative assessments is tabulated to highlight the value observed at the maximum user and network load. Therefore, the percentage of improvements is identified as a difference between cumulative sum of existing methods and the proposed method to distinguish the findings. The variations are highlighted in the graphical illustrations where the changes between the existing methods and the proposed are seen clearly. Besides, the percentage of improvement varies with the previously observed value and therefore minimal variations are observed. The comparative analysis results are tabulated in Tabs. 1 and 2 for the maximum users and network load. The final values with the difference between proposed and existing are used to identify the percentage of improvements for the users and network load.

The proposed RRP-MF reduces outage time by 13.79%, average delay by 13.18%, network overload by 14.04%, and computation complexity by 8.51%. This method improves the response ratio by 13.41% and resource utilization by 13.51%. This output is based on Tab. 1 considering the maximum number of users.

Table 1 Results of the metrics for users

Metrics	ERA-LA	RoISM	EFLSM	ESGCA	RRP-MF
Outage Time / s	1.319	1.014	0.901	0.705	0.5062
Average Delay / s	3.691	2.942	2.421	1.906	1.2544
Network Overload (Requests/ Interval)	52	40	27	21	12
Response Ratio	92.17	92.76	93.69	94.64	95.372
Resource Utilization	0.885	0.901	0.929	0.948	0.9782
Computation Complexity	0.191	0.15	0.145	0.124	0.1093

The proposed RRP-MF reduces outage time by 14.03%, average delay by 13.59%, network overload by 14.51%, and computation complexity by 8.7%. This method improves the response ratio by 13.64% and resource utilization by 13.55%. This output is computed based on the maximum network load output presented in Tab. 2.

Table 2 Results of the metrics for network load

Metrics	ERA-LA	RoISM	EFLSM	ESGCA	RRP-MF
Outage Time / s	1.328	1.128	0.902	0.774	0.5558
Average Delay / s	3.645	2.833	2.048	1.456	1.0232
Network Overload (Requests/ Interval)	53	37	28	16	11
Response Ratio	92.22	92.61	93.45	94.39	95.092
Resource Utilization	0.881	0.904	0.939	0.951	0.9745
Computation Complexity	0.192	0.154	0.132	0.116	0.099

5 CONCLUSION

The Reliable Resource Placement with Migration Function method ensures resilient and efficient service delivery in smart city networks. Optimal resource placement and suitable migration of services during outages or overloads enhance the efficiency of the communication network and wireless devices in SC. The process of resource projection through availability and allocation is based on migration. The incorporation of a transfer network with migration improves the experience of the end user and overall communication in smart cities. The proposed RRP-MF reduces outage time by 14.03%, average delay by 13.59% and network overload by 14.51%.

6 REFERENCES

[1] Nassereddine, M. & Khang, A. (2024). Applications of Internet of Things (IoT) in smart cities. *Advanced IoT technologies and applications in the industry 4.0 digital economy*, 109-136. https://doi.org/10.1201/9781003434269

[2] Ramal, P. J. & Anbalagan, E. (2026). Optimized deep learning model using capsule networks and GRUs for predictive analytics in smart cities. *Tehnički vjesnik - Technical Gazette*, *33*(2), 540-548. https://doi.org/10.17559/TV-20250410002574

[3] Adattil, R., Thorvald, P., & Romero, D. (2024). Assessing the psychosocial impacts of Industry 4.0 technologies adoption in the Operator 4.0: Literature review & theoretical framework. *International Journal of Industrial Engineering and Management*, *15*(1), 59-80. https://doi.org/10.24867/IJIEM-2024-1-348

[4] Li, H. & Xiao, X. (2025). Research on smart city governance and network management based on multi-source heterogeneous data fusion processing and analysis. *Tehnički vjesnik - Technical Gazette, 32*(3), 1162-1171. https://doi.org/10.17559/TV-20241209002176

[5] Vracar, L., Milovancevic, M., Vracar, J., Matijosius, J., Kilikevicius, A., & Buchmeister, B. (2026). MQTT enabled device for industrial environment and smart city. *Facta Universitatis, Series: Mechanical Engineering*, *24*(1), 171-182. https://doi.org/10.22190/FUME200901089

[6] Alshaflut, A. (2024). Enhancing smart city infrastructure: An iot-integrated system approach for real-time urban management. *Indian journal of science and technology*, *17*, 3012-3017. https://doi.org/10.17485/IJST/v17i29.1070

[7] Venumuddala, V. R., Prakash, A., & Chaudhuri, B. (2024). Governing smart city IoT interventions: A complex adaptive systems perspective. *Digital Government: Research and Practice*, *5*(3), 1-24. https://doi.org/10.1145/3660644

[8] Alwar, B., Edgardo, H., & Faustino, E. (2024). Integrating IoT and Artificial Intelligence for Sustainable Smart City Development: A Case Study Approach. *ITEJ (Information Technology Engineering Journals)*, *9*(2), 80-90. https://doi.org/10.24235/itej.v9i2.137

[9] Salih, S., Abdelmaboud, A., Husain, O., Motwakel, A., Elshafie, H., Sharif, M., & Hamdan, M. (2025). IoT in urban development: insight into smart city applications, case studies, challenges, and future prospects. *PeerJ Computer Science*, *11*, e2816. https://doi.org/10.7717/peerj-cs.2816

[10] Pavithra, N., Sapna, R., Manasa, C. M., & Devadas, R. M. (2025). Realization of Smart City Based on IoT and AI. In *IoT and Machine Learning for Smart Applications*, 127-142. https://doi.org/10.1201/9781032623276

[11] Singh, B., Jain, V., Kaunert, C., & Vig, K. (2024). Shaping highly intelligent Internet of things (IoT) and wireless sensors for smart cities. *Secure and Intelligent IoT-Enabled Smart Cities*, 117-140. https://doi.org/10.4018/979-8-3693-2373-1.ch007

[12] Taneja, A., Saluja, N., & Rani, S. (2024). An energy efficient dynamic framework for resource control in massive IoT network for smart cities. *Wireless Networks*, *30*(5), 4447-4458. https://doi.org/10.1007/s11276-022-03047-0

[13] Salam, M. H., Rathore, S. A., Maken, F. H., Janjua, J. I., & Irfan, M. (2025). Harnessing Big Data and IoT for Enhanced Resource Optimization in Sustainable Construction within Smart Cities. *The Journal of Research Review*, *2*(01), 90-104.

[14] Adaramola, T. S., Omole, O. M., Wada, I., Nwariaku, H., Arowolo, M. E., & Adigun, O. A. (2024). Internet of thing integration in green fintech for enhanced resource management in smart cities. *World Journal of Advanced Research and Reviews*, *23*(2), 1317-1327. https://doi.org/10.30574/wjarr.2024.23.2.2448

[15] Rahman, S., Islam, M., Hossain, I., & Ahmed, A. (2024). Utilizing AI and data analytics for optimizing resource allocation in smart cities: A US based study. *International journal of artificial intelligence*, *4*(07), 70-95.

[16] Darabkh, K. A. & Al-Akhras, M. (2025). A Novel Load-Driven Location-and Power-Aware EO-Based IoT-WSN Clustering and Routing Protocol for Sustainable Smart Cities. *IEEE Internet of Things Journal*, *1*(1), 99. https://doi.org/10.1109/JIOT.2025.3574636

[17] Dhiman, G. & Alghamdi, N. S. (2024). Smose: Artificial intelligence-based smart city framework using multi-objective and iot approach for consumer electronics application. *IEEE Transactions on Consumer Electronics*, *70*(1), 3848-3855. https://doi.org/10.1109/TCE.2024.3363720

[18] Sharma, S. K., AlGhamdi, R., Alasmari, S., Sharma, N. K., Khan, H., & Ahmad, F. (2024). Fractional order PID controllers for collaborative energy management in IoT-Smart cities: hybrid optimization algorithms for demand. *Energy Reports*, *12*, 5551-5566. https://doi.org/10.1016/j.egyr.2024.11.038

[19] Nikpour, M., Yousefi, P. B., Jafarzadeh, H., Danesh, K., Shomali, R., Asadi, S., ... & Ahmadi, M. (2025). Intelligent energy management with iot framework in smart cities using intelligent analysis: An application of machine learning methods for complex networks and systems. *Journal of Network and Computer Applications*, *235*, 104089. https://doi.org/10.48550/arXiv.2306.05567

[20] Wu, B., Ma, L., Ji, Y., Cong, J., Xu, M., Zhao, J., & Yang, Y. (2025). Multi-layer guided reinforcement learning task offloading based on Softmax policy in smart cities. *Computer Communications*, *235*, 108105. https://doi.org/10.1016/j.comcom.2025.10810

[21] Darabkh, K. A. & Al-Akhras, M. (2025). Towards optimized IoT sensor networks for smart cities: Centrality-Aware Position-Based Occlusion-Driven and role dynamics

solutions for clustering and routing. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2025.3570612

[22] Wang, T. & Zhao, Z. (2024). Implementing a smart city using internet of things based on wildwood combined with fractional order-based golden search algorithm. *Scientific Reports*, *14*(1), 29306. https://doi.org/10.1038/s41598-024-79972-1

[23] Naeem, M. A., Meng, Y., & Chaudhary, S. (2024). The Impact of Federated Learning on Improving the IoT-Based Network in a Sustainable Smart Cities. *Electronics*, *13*(18), 3653. https://doi.org/10.3390/electronics13183653

[24] Reddy, K. H. K., Srivastava, G., Goswami, R. S., & Roy, D. S. (2023). A hybrid optimized intelligent resource-constrained service scheduling for unified IoT applications in smart cities. *IEEE Transactions on Network and Service Management*, *21*(2), 1648-1659. https://doi.org/10.1109/TNSM.2023.3341296

[25] Reddy, K. H. K., Goswami, R. S., Luhach, A. K., Chatterjee, P., Alnumay, M., & Roy, D. S. (2024). Eflsm:-an intelligent resource manager for fog layer service management in smart cities. *IEEE Transactions on Consumer Electronics*, *70*(1), 2281-2289. https://doi.org/10.1109/TCE.2024.3376961

[26] Sahoo, S., Sahoo, K. S., Sahoo, B., & Gandomi, A. H. (2024). A learning automata based edge resource allocation approach for IoT-enabled smart cities. *Digital Communications and Networks*, *10*(5), 1258-1266. https://doi.org/10.1016/j.dcan.2023.11.009

[27] Wan, X. (2024). Dynamic resource management in MEC powered by edge intelligence for smart city internet of things. *Journal of Grid Computing*, *22*(1), 29. https://doi.org/10.1007/s10723-024-09749-3

[28] Adreani, L., Bellini, P., Fanfani, M., Nesi, P., & Pantaleo, G. (2024). Smart city digital twin framework for real-time multi-data integration and wide public distribution. *IEEE Access*, *12*, 76277-76303. https://doi.org/10.21203/rs.3.rs-3640569/v1

[29] Jeevanantham, S., Venkatesan, C., & Rebekka, B. (2024). Distributed neuro-fuzzy routing for energy-efficient IoT smart city applications in WSN. *Telecommunication Systems*, *87*(2), 497-516. https://doi.org/10.1007/s11235-024-01195-6

[30] Singh, A. R., Sujatha, M. S., Kadu, A. D., Bajaj, M., Addis, H. K., & Sarada, K. (2025). A deep learning and IoT-driven framework for real-time adaptive resource allocation and grid optimization in smart energy systems. *Scientific Reports*, *15*(1), 19309. https://doi.org/10.1038/s41598-025-02649-w

[31] Gou, F. & Wu, J. (2024). Optimization of edge server group collaboration architecture strategy in IoT smart cities application. *Peer-to-Peer Networking and Applications*, *17*(5), 3110-3132. https://doi.org/10.1007/s12083-024-01739-2

[32] Bangash, Y. A., Iqbal, W., Mussiraliyeva, S., Rubab, S., & Rauf, B. (2024). Reliability through an optimal SDS controller's placement in a SDDC and smart city. *Cluster Computing*, *27*(6), 7219-7240. https://doi.org/10.1007/s10586-024-04325-6

[33] Sun, Z., Yang, H., Li, C., Yao, Q., Teng, Y., Zhang, J., ... & Vasilakos, A. V. (2024). A resource allocation scheme for edge computing network in smart city based on attention mechanism. *ACM Transactions on Sensor Networks*. https://doi.org/10.1145/365003

[34] Cities, S. (7). AI-Optimized Urban Resource Management for Sustainable. *Financial Inclusion and Sustainable Development in Sub-Saharan Africa*, *96*. https://doi.org/10.4324/9781003515715

[35] Kapanski, A. A., Klyuev, R. V., Boltrushevich, A. E., Sorokova, S. N., Efremenkov, E. A., Demin, A. Y., & Martyushev, N. V. (2025). Geospatial clustering in smart city resource management: An initial step in the optimisation of complex technical supply systems. *Smart Cities*, *8*(1), 14. https://doi.org/10.3390/smartcities8010014

[36] Papaioannou, C., Dimara, A., Papaioannou, A., Tzitzios, I., Anagnostopoulos, C. N., & Krinidis, S. (2025). Hierarchical Resources Management System for Internet of Things-Enabled Smart Cities. *Sensors*, *25*(3), 616. https://doi.org/10.3390/s25030616

[37] Singh, K. D., Singh, P. D., Verma, R., & Maurya, S. (2024, July). Optimizing urban resource management through cloud and fog computing in smart cities. *AIP Conference Proceedings*, 3121(1), 030002. https://doi.org/10.1063/5.0222009

[38] Peshattiwar, A., Prasad, A. B., Misra, S., Painuly, S., & Deepti, K. (2025). A Novel Approach for Enabling Service Migrations for IoT Smart City Applications: Architecture for Service Management. In *Technological Applications for Smart Sensors* (pp. 521-538). Apple Academic Press.

[39] Mohamed, M., & Alosman, K. (2024). An IoT-enabled framework for smart city infrastructure management. *Authorea Preprints*. https://doi.org/10.36227/techrxiv.170906134.46169506/v1

[40] Bellini, P., Bologna, D., Nesi, P., & Pantaleo, G. (2024). A Unified Knowledge Model for Managing Smart City/IoT Platform Entities for Multitenant Scenarios. *Smart Cities*, *7*(5), 2339-2365. https://doi.org/10.3390/smartcities7050092

[41] https://ieee-dataport.org/documents/iot-time-series-traffic-data-smart-city-ehealth-and-smart-factory#files

Contact information:

Shanmugham M.
(Corresponding author)
Department of Electronics and Communication Engineering,
Builders Engineering College,
Kangayam - 638108, Tamilnadu, India
Email: shanmugampdhd5@gmail.com

Chandrasekaran V., Professor
Department of Electronics and Communication Engineering,
Velalar College of Engineering and Technology,
Erode - 638-12, Tamil Nadu, India
Email: vc4sachin@gmail.com