UDC 004.41:519.7
DOI: 10.15587/1729-4061.2025.344553

DEVISING A METHOD FOR MANAGING COMPUTING RESOURCES IN A FOG LAYER OF THE MOBILE HIGH-DENSITY INTERNET OF THINGS

Heorhii Kuchuk
Doctor of Technical Sciences, Professor*
Oleksandr Mozhaiev
Corresponding author
Doctor of Technical Sciences, Professor**
E-mail: mozhaev1957@gmail.com
Serhii Tiulieniev
PhD
Director
Scientific Research Center for Forensic Expertise in the Field of Information Technologies and Intellectual Property of the Ministry of Justice of Ukraine
L. Ukrainka blvd., 26, Kyiv, Ukraine, 01133
Mykhailo Mozhaiev
Doctor of Technical Sciences, Senior Researcher, Acting Director****
Nina Kuchuk
Doctor of Technical Sciences, Professor*
Pavlo Khorobrykh
PhD, Acting First Deputy Director****
Yurii Gnusov
PhD, Associate Professor**
Yurii Horelov
PhD, Associate Professor**
Vitalii Svitlychnyi
PhD, Associate Professor
Department of Information Systems and Technologies***
Oleksandr Bilyk
PhD Student
V. V. Popovskyy Department of Infocommunication Engineering
Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166
*Department of Computer Engineering and Programming
National Technical University "Kharkiv Polytechnic Institute"
Kyrpychova str., 2, Kharkiv, Ukraine, 61002
Department of Cyber Security and DATA Technologies*
***Kharkiv National University of Internal Affairs
L. Landau ave., 27, Kharkiv, Ukraine, 61080
****Dnipropetrovsk Research Institute
of Forensic Examinations of the Ministry of Justice of Ukraine
Sicheslavska Naberezhna str., 17, Dnipro, Ukraine, 49000

This study considers a process that manages the distribution of computing resources in the fog layer of the mobile high-density Internet of Things. The task addressed is to reduce the load imbalance of fog servers by devising a method for controlling computing resources in the fog layer when processing information flows.

Information flows are formed by intelligent gateways of the mobile high-density Internet of Things, which receive data from the boundary layer. In the process of research, a mathematical model for the process of controlling computing resources in the fog layer was built. Its main difference from existing ones is a module hierarchical structure according to the basic levels of decision-making when managing computing resources.

When constructing the model, the principle of process decomposition into adjacent time intervals was used. Its application made it possible to carry out local optimization of the process of managing computing resources in short time intervals. The mathematical model has made it possible to devise a method for controlling computing resources in the fog layer.

The main difference of this method from existing ones is that the process optimization is carried out according to the area of the relative deviation from the balanced load in the time interval under study. In addition, a two-stage algorithm for distributing tasks of free fog devices across fog layer servers is also used. That made it possible to reduce the time for finding an approximate solution for distributing computing resources of fog servers by up to 50%.

The research results can be attributed to the combined use of the simulated annealing algorithm and the genetic algorithm. The method is effective when the load on the fog layer is from 20% to 70% of the maximum possible load

Keywords: mobile device, gateway, simulated annealing algorithm, genetic algorithm, IoT ecosystem

Received 12.09.2025
Received in revised form 12.11.2025
Accepted 21.11.2025
Published 17.12.2025

How to Cite: *Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Khorobrykh, P., Gnusov, Y., Horelov, Y., Svitlychnyi, V., Bilyk, O. (2025). Devising a method for managing computing resources in a fog layer of the mobile high-density internet of things. Eastern-European Journal of Enterprise Technologies, 6 (4 (138)), 15–25.*
https://doi.org/10.15587/1729-4061.2025.344553

1. Introduction

The rapid evolution of Internet of Things (IoT) technologies has led to significant changes in approaches to data collection, transmission, and processing [1]. State-of-the-art IoT systems combine billions of interconnected devices that provide interaction between physical objects and information infrastructures [2]. Such integration helps increase the efficiency of process management in many areas [3, 4].

One of the fields of IoT development is the Mobile High-Density Internet of Things (MHDIoT). It is characterized by a significant concentration of mobile devices within limited geographical areas [5]. This system provides connectivity of a large number of IoT devices under limited space conditions.

However, MHDIoT is characterized by uneven traffic, dynamic changes in the topology of the support ecosystem network, as well as high requirements for data exchange speed [6]. The characteristic features of MHDIoT are the combination of mobility and high density of sensors with high heterogeneity of devices and protocols [7]. In addition, the MHDIoT support ecosystem must be able to process large amounts of information in real time. At the same time, under conditions of rapidly changing network characteristics, it is necessary to enable connection stability, energy efficiency, and specified quality of service (QoS) parameters [8].

The MHDIoT ecosystem operation is accompanied by a number of technical problems. The main problematic issues are the following [9]:

– overload of communication channels;

– limited computing resources at peripheral levels;

– the complexity of effective management of routing of information flows in the MHDIoT support ecosystem.

In addition, the high dynamics of device movement complicate the processes of load balancing, protocol coordination, and maintaining network stability.

To increase the efficiency of MHDIoT functioning under such conditions, a fog layer is widely used [10]. It acts as an intermediate link between cloud services and peripheral devices [11]. Fog nodes and servers perform local data processing [12]. This allows for the following:

– reducing delays in data transmission and processing processes;

– reducing the load on MHDIoT cloud data centers;

– implementing a real-time mode for processing operational transactions;

– increasing the overall performance of the MHDIoT support ecosystem.

For mobile devices in the fog layer, the first issue to consider is energy efficiency. In addition, it should be noted that the computing resources of devices and servers in the fog layer are limited [13]. Also, any fog layer is a distributed system, which consists of a number of dynamic mobile clusters with centralized management. Typically, in the MHDIoT fog layer there is always a large number of individual mobile devices that do not belong to any cluster. Such devices contribute to generating an imbalance in the use of computing resources.

Therefore, the issue of improving the efficiency of managing computing resources in the fog layer is relevant. Solving this task could make it possible to increase the functioning efficiency of the MHDIoT support ecosystem.

2. Literature review and problem statement

In [14], an approach to managing the process of information flow in the IoT ecosphere is proposed. In particular, the construction of virtual clusters at the edge of the Internet of Things is proposed. Each cluster is oriented to a specific gateway that maintains communication with the corresponding clusters of the fog layer. This approach allows for the effective use of computing resources of the boundary and fog layers of the IoT support ecosystem. However, the mobility of a portion of devices is not taken into account, which does not allow the formation of stable static clusters on both layers. In addition, the high density of IoT devices contributes to the overloading of the ecosystem in some areas due to restrictions on the number of elements of virtual clusters.

Such restrictions are removed in [15], that is, the algorithm proposed in it could be used for high-density IoT networks. However, as in [14], the designed virtual clusters will not be stable because of the mobility of most devices. The features of mobile high-density IoT are taken into account in [16]. The work proposes an MHDIoT architecture that takes into account the specificity of the system. However, the limitation of the proposed algorithm regarding the mandatory clustering of all devices of the boundary and fog layers is usually impossible to implement in real systems. Due to mobility, it is impossible to bind a fraction of fog devices to a specific fog cluster. Such free devices establish a connection with the nearest fog cluster during periods of activity, which leads to an imbalance in the load on fog servers. As a result, delays in servicing tasks increase. Therefore, when managing the computing resource of the fog layer, it is necessary to take into account the directions of requests from free fog devices.

Balancing algorithms for IoT systems are proposed in many studies. Thus, in [17], the study is aimed at achieving load balancing using the method of optimal server selection and resource allocation based on TOPSIS. TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is a multi-criteria decision-making method. It determines the best alternative, the closest to the ideal solution and the farthest from the worst. The method is based on the normalization of criteria and determining the ranking of alternatives by the degree of proximity to the optimal option. However, the process of obtaining the required solution takes time, which is unacceptable for MHDIoT.

The method of load balancing for IoT/Fog/Cloud environments, proposed in [18], provides faster proposals. The method is based on the prediction of the workload and the presence of unstable mobile nodes and could be used for decentralized systems with a high density of elements. However, the method does not take into account the presence of dynamic mobile clusters. Therefore, when distributing computing resources, it is impossible to distinguish between free and cluster fog nodes.

The balancing algorithm for IoT systems proposed in [19] also works quickly. However, balancing occurs by switching to less congested channels, without taking into account the mobility of fog devices. The mobility of fog devices is taken into account in [20] but as in [16], the specificity of free fog devices are not taken into consideration.

In [21], the authors proposed task distribution by implementing the Hopcroft-Karp algorithm, which combines breadth-first and depth-first search. This algorithm reduces the distribution delay and improves the quality of service. However, a mandatory requirement is full clustering of fog devices, which cannot be performed in MHDIoT.

Therefore, there are reasons to argue that it is advisable to conduct research aimed at reducing the load imbalance at fog servers when processing information flows of mobile high-density IoT.

3. The aim and objectives of the study

The aim of our study is to devise a method for managing fog layer computing resources when processing information flows generated by mobile high-density Internet of Things gateways. This will make it possible to meet the quality of service (QoS) requirements even with a high density of mobile devices by reducing the load imbalance of fog servers.

To achieve the goal, the following tasks were set:

– to construct a mathematical model of the fog layer computing resource management process;

– to define the criterion and quality indicators of the fog layer computing resource management process;

– to develop a two-stage algorithm for distributing tasks of free fog devices across fog layer servers.

4. The study materials and methods

The object of our study is the process of managing the distribution of computing resources in the fog layer of the mobile high-density Internet of Things. The work considers mobile fog devices that have accepted tasks generated by MHDIoT gateways for transmission to fog servers for further processing. The mobile fog device selects for transmission one of the accessible fog servers that has sufficient computing resources for further processing.

The principal hypothesis of the study assumes that the implementation of a new method for managing computing resources in the fog layer could make it possible to reduce the imbalance of the fog server load. The method is based on finding an approximate solution to a combinatorial optimization problem of high dimensionality. This will enable increased efficiency in the use of limited computing resources in the fog layer, and accordingly, the efficiency of MHDIoT operation will increase.

When devising the method for managing computing resources, the following conditions were used:

Condition 1. A foggy mobile device enters the active state immediately after receiving a task for processing by the foggy server.

Condition 2. Gateways accept tasks from edge layer devices and transfer them to foggy devices as they accumulate, in discrete time intervals.

Condition 3. Devices of temporary persistent foggy mobile clusters transfer tasks only to the server that is the center of the corresponding cluster.

Condition 4. The free computing resource of each foggy server is calculated only after servicing all active elements of the corresponding cluster.

Condition 5. Each persistent foggy mobile cluster has only one foggy server, which is the center of this cluster.

In the process of devising a method for managing the distribution of computing resources in the foggy layer of the mobile high-density Internet of Things, a number of different methods and algorithms were applied.

When constructing a mathematical model of the process that manages the computational resources in the fog layer, the principle of decomposition of the process into adjacent time intervals was used [22]. The principle implies dividing the time interval of the studied process into a sequence of smaller time intervals that have a lower computational complexity in the research process [23]. Instead of considering the system or process as a whole for a long time, the entire time range $[0; T]$ is divided into K adjacent intervals in the following way

$$[0; T] = [t_0; t_1] \cup [t_1; t_2] \cup \ldots \cup [t_{K-1}; t_K], \tag{1}$$

where $t_0 = 0$; $t_K = T$; $[t_{k-1}; t_k]$ is a separate time subinterval, $k = 1 \ldots K$.

At each interval, the process is described by its own submodel. Such submodels may have different parameters or external conditions but the connections between the intervals are preserved through boundary conditions

$$x\left(t_k^-\right) = x\left(t_k^+\right), \tag{2}$$

that is, the final state of the previous interval is the initial state of the next.

The principle of time decomposition is used for:

– simplification of calculations or optimization of complex dynamic processes;

– increasing the accuracy of numerical modeling;

– construction of adaptive or recurrent control algorithms;

– organization of parallel calculations (when different time intervals can be processed independently);

– local optimization of control over short time intervals.

In particular, when controlling the distribution of computational resources in the MHDIoT fog layer, the process can be performed not continuously, but in time slots. For each interval, local optimization of resources is performed, after which the results are transferred to the next interval as initial conditions.

Therefore, the application of this principle in constructing a hierarchical model of a dynamic process provides the following advantages:

– reduction of computational complexity;

– flexibility in changing model parameters at different stages;

– possibility of adaptive or online control;

– convenience for parallel or distributed implementation.

When determining the quality indicators of the fog layer computational resource management process, the concept of the area of the time process parameter S was used [24]. The value of S is an integral characteristic of a dynamic process, which shows the accumulated value of a certain parameter over time.

Let us consider some continuous time process $x(t)$, which determines the change of some system parameter over time t. Then its area on the time interval $[t_0; t_1]$ is defined as

$$S_{continuous} = \int_{t_0}^{t_1} x\left(t\right) \mathrm{d}t. \tag{3}$$

In the discrete case, such as when using partitioning (1), the area of the discrete parameter is calculated over the boundaries of discrete intervals as follows

$$S_{discrete} = \sum_{i=t_0}^{t_1} x\left(t_i\right) \cdot \Delta t_i. \tag{4}$$

When analyzing the load on the MHDIoT fog layer: the area of the parameter can be used, for example, to estimate the total load on the nodes for a certain period of time.

When finding solutions to the combinatorial optimization problem by a two-stage method, the Simulated Annealing Algorithm [25] was used at the first stage. The algorithm uses a stochastic metaheuristic method to find an approximate solution to the combinatorial problem. The method simulates the physical annealing process: the system cools down gradually, sometimes taking "worst" states to avoid local minima. The algorithm can be described by the following sequence of actions:

1. Initialization of the simulated annealing process. The initial values of the algorithm variables are set:

– initial solution x_0;

– initial temperature T_0;

– number of iterations N at one temperature level.

2. Main cycle. For each iteration, a new solution x_{new} is generated that is a neighbor to current solution x. The change in the quality function (fitness function) is calculated as follows

$$\Delta E = f(x_{new}) - f(x). \tag{5}$$

3. Acceptance criterion. If $\Delta E < 0$, that is, the result is improved, then we accept the new solution x_{new} as the current solution. Otherwise, the new value is accepted with a probability calculated as

$$P_{accept} = e^{-\Delta E/T}. \tag{6}$$

4. Cooling. The temperature decreases according to the following scheme

$$T_{k+1} = \alpha \cdot T_k,\ 0 < \alpha < 1, \tag{7}$$

where T_k is the temperature value at the k-th step; α is the cooling rate, which determines how quickly the system temperature decreases with each iteration.

If α is close to 1 (for example, it takes the value 0.95...0.99), then the cooling is slow, the algorithm has a better chance of finding a global minimum, but it works longer. If α is less than 0.8, then we have fast cooling, the algorithm works faster but can "get stuck" in a local minimum.

5. Termination. The process is repeated until temperature T becomes very small or the maximum number of iterations is reached. The process can also be terminated if there is no improvement in the solution for a certain number of steps.

The given algorithm is simple and universal; it works for continuous and discrete problems. Due to stochastic deviations, the algorithm avoids local minima. By choosing the value of the cooling rate, the algorithm allows one to speed up the process of finding a solution, which is essential for the MHDIoT ecosystem.

In the second stage of the two-stage method, the selection of a new population was carried out using a classical genetic algorithm [26].

A genetic algorithm is an evolutionary optimization method that imitates natural selection: the best solutions "survive" and combine to form new generations [27]. The algorithm can be described by the following sequence of actions:

1. Initialization of the algorithm. An initial population is created

$$P_0 = \{x_1, x_2, ..., x_n\}, \tag{8}$$

where each element of the given set is formed by the corresponding workflow of the first stage and is a variant of the distribution of computational resources.

2. Fitness assessment. For each i-th element of the current population, the value of the fitness function or fitness function is calculated: $f_i = f(x_i)$.

3. Selection. Individuals are selected for reproduction in proportion to their fitness, for example, there may be such a selection operator

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i}, \tag{9}$$

where p_i is the probability of selecting individual x_i.

4. Crossover or crossing. Offspring are created from two selected parents

$$x_{new} = (1-\beta)\cdot x_a + \beta \cdot x_b, \tag{10}$$

where x_a and x_b are the chosen parents, and β is the crossover parameter, $0 < \beta < 1$.

5. Mutation. A random change is made to some elements of the individuals according to the formula

$$x_{new} = x_i + \delta,\ \delta \sim N(+0, \sigma^2), \tag{11}$$

that is, mutation parameter δ is distributed according to the normal distribution law near 0 with variance σ^2 and has only non-negative values.

6. Formation of a new generation. Steps 2–5 are repeated until the stopping criterion is met, that is, until the maximum iterations or the specified accuracy is reached.

So, the key formula for population update in the classical genetic algorithm takes the following form

$$P_{k+1} = \text{Mutation (Crossover (Selection }(P_k))). \tag{12}$$

Using a classical genetic algorithm at the second stage of searching for an approximate solution allows one to speed up the annealing process by correcting the initial population of individuals.

5. Results related to devising and investigating a method for managing computing resources in the MHDIoT fog layer

5. 1. Mathematical model of the process that controls computing resources in the fog layer

The ecosystem of the mobile high-density Internet of Things has a specific four-layer architecture. The lower layer of the ecosystem contains IoT devices that are engaged in information collection. The next, boundary layer, is focused on receiving and preliminary composition of information from IoT devices. Both individual mobile elements and IoT devices can be used as devices of this layer. Such IoT devices must have the ability to simultaneously receive information from several sensors, accumulate it and transmit it to the gateway for communication with the fog layer. A feature of boundary layer devices is their location at a short distance from the lower layer devices, which provides the ability to receive information in real time.

Communication between the boundary and fog layers is carried out using intelligent gateways. They act as an intermediary between edge devices and fog nodes, providing data processing and transmission at a short distance from the source of their origin.

At the same time, gateways are able to perform a number of simple functions, such as:

– local data preprocessing, that is, minimal data analysis and aggregation before further transmission;

– information filtering and compression to reduce the amount of transmitted data to save channel bandwidth;

– information security and encryption to control access, authenticate devices, and protect communication channels;

– data conversion between different communication protocols;

– making operational decisions in emergency situations without the need to refer to higher layers.

Based on the above tasks, intelligent gateways of the mobile high-density Internet of Things ecosystem should have the following characteristics:

– the presence of built-in machine learning or analytics algorithms;

– the presence of support for several types of connections;

– the ability to connect a large number of boundary layer devices;

– the ability to perform some data processing functions in the absence of a connection to the fog layer;

– energy efficiency due to optimal energy consumption.

The fog layer will receive IoT data streams generated by intelligent gateways. This layer is an intermediate, "intelligent" link between edge devices and the cloud.

Elements of the fog layer are local servers, stationary and local fog nodes, micro data centers located close to IoT data sources. They provide distributed processing, storage, and management of information flows in real time and perform a number of functions, the main of which are the following:

– local (intermediate) processing of data received from gateways;

– the ability to respond in real time to extreme events;

– temporary data storage (caching);

– the ability to work with limitations even when there is a temporary lack of communication with the cloud;

– reduce the amount of data transmitted to the cloud;

– store critical information locally without sending it to the cloud;

– minimize the amount of data transmitted further through filtering or compression;

– enable stable system operation even with limited network bandwidth.

Among the elements of the fog layer, the main work is performed by fog servers. They maintain constant communication with the cloud data processing center, transmitting information flows to the cloud and receiving the results of the analysis. In addition, the MHDIoT fog layer servers are the centers of mobile clusters that are stable for a fixed time interval *T*. The components of such clusters are mobile components of the fog layer, which during this interval are constantly in the coverage area of the corresponding fog server. These components have a small computing resource. The main task of these components is to accept information flows generated by the intelligent gateway, perform minimal processing if necessary and transfer them to the fog server.

But in the MHDIoT fog layer there are still many free mobile elements that do not belong to any cluster. Such elements, if necessary, transfer information, look for the nearest fog server.

The cloud layer is the top layer of the MHDIoT ecosystem, which provides centralized storage, processing, analytics, and data management from the entire MHDIoT environment. Its role is to perform tasks that require high computing power, scalability, and global access.

The cloud layer receives data from fog and edge devices and stores it in databases, Big Data repositories, or data centers. Cloud services perform Big Data analytics, AI/ML processing, pattern recognition, and event prediction. The cloud also provides centralized administration of IoT devices. But all tasks that require real-time or near-real-time operations must be performed closer to the data, that is, in the fog layer.

So, the MHDIoT ecosystem has the architecture shown in Fig. 1. Fig. 1 shows the following:

– Things – the device layer of the Internet of Things;

– MD A – an active mobile device of the boundary layer;

– MD N – an inactive mobile device of the boundary layer;

– GW – an intelligent gateway between the boundary and fog layers;

– FCA – an active mobile fog device that is part of a cluster;

– FCN – an inactive mobile fog device that is part of a cluster;

– FA – a free active mobile fog device;

– FN – a free inactive mobile fog device.

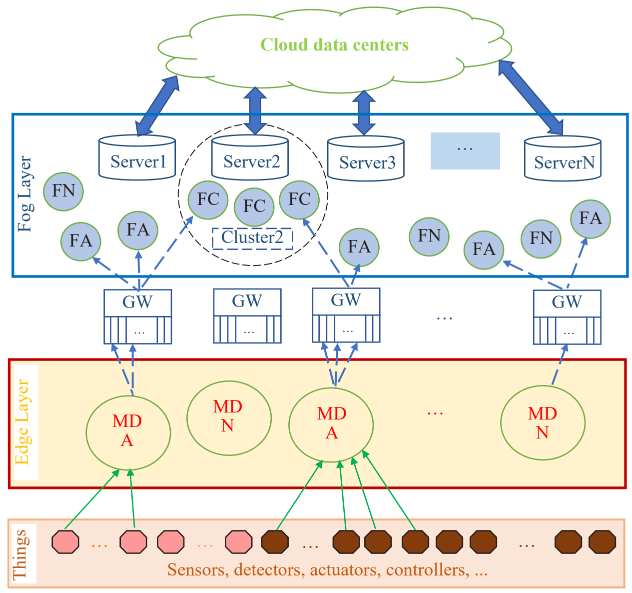

Fig. 1. Example of mobile high-density IoT ecosystem architecture

The boundary layer collects raw data from sensors. Intelligent gateways pre-process, filter, and transmit data to the fog layer. The cloud layer is responsible for global analytics, long-term storage, and strategic management. The fog layer is responsible for operational tasks, local analytics, security, and information transmission to the cloud layer. In MHDIoT, the fog layer usually carries the most workload. The fog layer's computational resource control system (CSCR) has a hierarchical structure. To build a model, the CSCR is divided into hierarchical decision-making levels. Each level reduces the uncertainty of a complex situation by defining and fixing a number of parameters for the higher level. Four basic decision-making levels are distinguished, shown in Fig. 2.

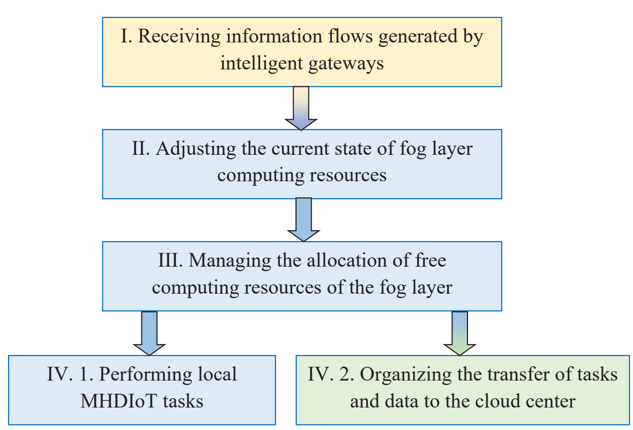

Fig. 2. Basic levels of decision-making in managing fog computing resources

The first-level programs organize the reception of data from the boundary layer. The second-level programs are components of the monitor of the current state of the computing resources in the fog layer. The third level provides the distribution of free computing resources. At this level, decisions are made related to the management of the load of servers of the fog layer clusters. These decisions are determined by the priorities of the operating modes, the task queue, the state of system resources, operator directives, etc. At the fourth level, fog servers perform local tasks of the Internet of Things and organize the transmission of information flows to the cloud data center.

Let CSCR be represented by a set of modules S_i, $i=1..4$, where i is the corresponding basic level of the hierarchy, as shown in Fig. 1. The fog layer has K fog servers, which make up the set $MK=\{Cl_1, \ldots, Cl_k, \ldots, Cl_K\}$, card $MK=K$. In addition, the components of the fog layer are also M_Σ fog devices that accept tasks from gateways. Each fog server is the center of a stable fog cluster.

The model considers a time interval of length T of conditional time units, which is a union of I disjoint intervals $\tau_i=[t_i, t_{i+1}]$

$$T=\bigcup_{i=0}^{I-1}\tau_i=\bigcup_{i=0}^{I-1}\left[t_i;t_{i+1}\right], \tag{13}$$

where t_0 is the start time of time interval T; t_i is the start time of the i-th partition interval, t_I is the end time of time interval T; $\Delta t_i=(t_{i+1}-t_i)$ is the size of the i-th partition interval.

At the beginning of each i-th interval, the current state of the fog layer is monitored. During this interval, the state of the fog devices does not change.

Let the following parameters be obtained as a result of monitoring time interval τ_i:

– M_i is the number of free devices, that is, those that are not involved in fog clusters, $M_i<M_\Sigma$;

– M_{ia} is the number of active free devices, that is, those that received tasks from the gateways of the combination of the boundary and fog layers, $M_{ia}\le M_i$;

– r_{iak} is the size of the computing resource that an active free device a needs to transfer the task to the fog server k, $a=1\ldots M_{ia}$, $k=1\ldots K$;

– r_{ik} is the size of the available computing resource on server k, which will remain after servicing the tasks of its own cluster, $k=1\ldots K$;

– r_{iak} is the size of the generalized costs of transferring tasks from the fog device a to server k.

If the server k during the current time interval does not have sufficient computing resources, then $r_{ik}=0$.

Active free devices of time interval τ_i constitute the set $MIA=\{m_{i1}, \ldots, m_{ia}, \ldots, m_{iMia}\}$, card $MIA=M_{ia}$. The distribution of the free resource of fog servers occurs according to one of the possible options γ_i for time interval τ_i

$$\gamma_i=\{(m_{ia}, k)\},\ a=1..M_{ia},\ k=1..K, \tag{14}$$

where $m_{ia}\in MIA$ and the following mandatory conditions are met:

1) each active fog device can choose only one fog server to transmit the task, that is

$$m_{ia_1}=m_{ia_2}\Rightarrow a_1=a_2; \tag{15}$$

2) all tasks received from IoT edge layer gateways must be transferred to fog servers, that is

$$\text{card }\gamma_i=M_{ia}; \tag{16}$$

3) the free resource of each fog server should be enough to implement tasks transferred from fog devices, that is

$$\sum_{\left(\ell\,|\,(m_{i\ell},k)\in\gamma_i\right)} r_{i\ell k}<r_{ik},\ k=1..K. \tag{17}$$

To select the required resource allocation option γ_i^*, it is necessary to form the corresponding *Crit* criterion in such a way that the following condition is met

$$\gamma_i^*=extr\left(\underset{\{\gamma_i\}}{Crit}\left(\gamma_i\right)\right). \tag{18}$$

With the selected control quality criterion, expression (18) will be the objective function of the combinatorial optimization problem. Accordingly, conditions (15) to (17) will be considered as a constraint to this problem.

5. 2. Criteria and quality indicators of the fog layer computing resource management process

The quantitative characteristics of CSCR should include quality indicators measured in one way or another, which include the following:

– average queue length, which determines the average number of tasks in the queue for the analyzed period;

– useful load of fog servers;

– fog layer throughput as the number of tasks served per time period;

– load balance as the uniform load of all fog servers;

– "honesty" in relation to CSCR to tasks as the variance of the waiting time of tasks in the queue;

– completeness of coverage as the ratio of the number of tasks processed before the target deadline to the total number of tasks.

All of these quality indicators primarily reflect the efficiency of CSCR distribution. In addition, quality indicators that reflect the scalability of CSCR can be used:

– the maximum number of tasks that can be served by fog layer servers;

– the maximum number of tasks that can be simultaneously sent by MHDIoT ecosphere gateways.

Next, the most commonly used quality indicators of the fog layer computing resource management system of the MHDIoT ecosystem are analyzed.

In the time interval $\tau_i=[t_i; t_{i+1}]$ at time t_i, MHDIoT gateways activate fog devices, transmitting information received from the boundary layer. Fog devices form two types of tasks for fog servers:

– tasks of the first type that need to be transmitted to the cloud layer;

– tasks of the second type that need to be performed on the fog server.

Tasks of the first type, transferred from device a to server m at interval τ_i, require r_{iam1} conditional units of computing resource. Tasks of the second type under such conditions require r_{iam2} conditional units. Then in general, the server must have no less than $r_{iam}=r_{iam1}+r_{iam2}$ conditional units of computing resource. It should be taken into account that all active fog devices belonging to fog clusters transfer tasks to the server of this cluster.

Free fog devices can choose any fog server that has sufficient computing resource left.

Let the task of fog device a be queued to fog server k at time t_{iak1}. The task is accepted for processing at time t_{iak2} and completed at time t_{iak3}. Then the time $P_{iak} = t_{iak1} - t_i$ will be the time of delivery of the task to the fog server. The time $Q_{iak} = t_{iak2} - t_{iak1}$ will be the waiting time of task a in the queue, and the time $E_{iak} = t_{iak3} - ti_{ak2}$ will be the execution time of task a. Therefore, the walltime of the task in the system is defined as

$$WT_{iak} = P_{iak} + Q_{iak} + E_{iak}.. \tag{19}$$

To reduce time WT_{iak}, it is necessary to minimize the components of formula (19).

The average waiting time of a task in the queue $Q(T)$ for time interval T is called the slowdown coefficient. It can be calculated as

$$Q(T) = \sum_{i=1}^{I}\sum_{a=1}^{M_{ia}}\sum_{k=1}^{K}\frac{Q_{iak}}{K}. \tag{20}$$

It should be noted that in real systems the spread of $Q\ i$ values can be significant. Therefore, the median value of this indicator is often used instead of the average waiting time of a task in the queue. Therefore, the average waiting time of a task in the queue has ceased to be an informative indicator and can only compare the quality of different resource management systems for the same information flow.

The time WT of a task in the system is some general indicator that characterizes the quality of the allocation of fog layer resources. This affects both the waiting time of a task in the queue and the performance of fog servers.

During each separate time interval τ_i, the values of the main numerical indicators of the fog layer state do not change. In particular, the total available free resource of the fog layer is

$$R_i = \sum_{k=1}^{K} r_{ik},\ i \in \overline{1,I}. \tag{21}$$

Then the time area of the available resource will be

$$R_\Sigma = \sum_{i=1}^{I} R_i \cdot \Delta t_i = \sum_{i=1}^{I}\sum_{k=1}^{K} r_{ik} \cdot \Delta t_i. \tag{22}$$

The total required resource of the fog layer of time interval τ_i is

$$r_i = \sum_{a=1}^{M_{ia}}\sum_{k=1}^{K} r_{iak},\ i \in \overline{1,I}. \tag{23}$$

The relative time area of the required resource is calculated as follows

$$r_\Sigma = \sum_{i=1}^{I} r_i \cdot \Delta t_i = \sum_{i=1}^{I}\sum_{a=1}^{M_{ia}}\sum_{k=1}^{K} r_{iak} \cdot \Delta t_i. \tag{24}$$

The load of computational resources of the fog layer during time interval T is calculated as follows

$$U(T) = \frac{r_\Sigma}{R_\Sigma} = \frac{\sum_{i=1}^{I}\sum_{a=1}^{M_{ia}}\sum_{k=1}^{K} r_{iak} \cdot \Delta t_i}{\sum_{i=1}^{I}\sum_{k=1}^{K} r_{ik} \cdot \Delta t_i},\ \bigcup_{i=1}^{I} \Delta t_i = T. \tag{25}$$

In some cases, the downtime rate of computing resources $I(T)$ is used as the opposite of the load

$$U\_(T) = 1 - U(T). \tag{26}$$

At a fixed load, the downtime will be minimal with a balanced distribution of fog layer resources.

At the i-th time interval, the relative free resource for the k-th fog server will be

$$\xi_{ik} = \frac{r_{ik}}{R_i},\ \sum_{k=1}^{K}\xi_{ik} = 1. \tag{27}$$

When choosing the distribution option γ_i, free fog devices transmit to the k-th fog server tasks that require the following relative amount of computing resources

$$\zeta_{ik} = \frac{\sum_{a=1}^{M_{ia}} r_{iak}}{r_i};\ \sum_{k=1}^{K}\zeta_{ik} = 1. \tag{28}$$

Then the normalized relative deviation from the balanced load of fog layer servers at the i-th time interval is calculated as

$$\eta_i = \sum_{k=1}^{K}\frac{\left(\xi_{ik} - \zeta_{ik}\right)^2}{K}. \tag{29}$$

Now it is possible to calculate the normalized area of the relative deviation from the balanced load over the entire considered interval of length T

$$\eta = \sum_{i=1}^{I}\frac{\eta_i \cdot \Delta t_i}{T}. \tag{30}$$

The minimum value of this indicator will indicate the best balance of the distribution of resources of the fog layer, and as a result, a decrease in deceleration and downtime coefficients.

5. 3. Two-stage algorithm for distributing tasks of free fog devices to fog layer servers

The task of distributing unoccupied fog layer resources is equivalent to the task of mapping free fog devices to fog servers.

We consider the set of fog servers $MK = \{Cl_1, \ldots, Cl_k, \ldots, Cl_K\}$ and free fog devices $MIA = \{m_{i1}, \ldots, m_{ia}, \ldots, m_{iMia}\}$ in time interval τ_i. Each resource distribution option γ_i corresponds to a functional, everywhere defined on MIA correspondence between the sets MIA and MK

$$\text{Y}: MIA \to MK. \tag{31}$$

In the general case, this correspondence is neither injective nor surjective. Its graphical representation is a bigraph G. The vertices of this graph are weighted elements of the sets MIA and MK. The weight of each vertex is the corresponding size of the computational resource. Next, we consider the sequence of variants γ_i, given by vector γ

$$\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_i, \ldots, \gamma_I\}. \tag{32}$$

Then, using (30), it is possible to formulate the objective function of the optimization problem for the distribution of resources of the for layer in the IoT ecosphere as follows

$$\eta \xrightarrow[\{\gamma\}]{} \min. \tag{33}$$

In this case, it is necessary to take into account a number of restrictions given by expressions (15) to (17).

Most of the input data of this optimization problem is formed on the basis of a short-term forecast. The main requirement for its solution is the time to obtain at least an approximate solution. Therefore, the use of exact methods for finding a solution is impractical. Reducing the solution time to acceptable values is possible by using approximate algorithms. These algorithms generally find a rational solution close to the optimal one, which provides the minimum value of the objective function (33). Among the approximate algorithms, heuristic algorithms are distinguished, which can be divided into iterative and population. Population algorithms include genetic algorithms, swarm algorithms, and evolutionary algorithms. According to the criterion of the speed of finding, among the iterative algorithms, the following can be distinguished:

– Tabu Search Algorithm, which uses the memory of previous solutions to avoid local minima;

– Greedy Randomized Adaptive Search Algorithm (GRASA);

– Simulated Annealing algorithm, the search speed depends significantly on the initial settings.

Among the heuristic algorithms, the most widespread are the simulated annealing and genetic selection algorithms. In complex combinatorial problems, the genetic algorithm outperforms other heuristics in terms of solution accuracy, while the simulated annealing algorithm is one of the fastest algorithms. Therefore, the simulated annealing algorithm was chosen to find a solution to the optimization problem.

The function for calculating the normalized area of the relative deviation was chosen as the fitness function. To speed up the solution finding, a two-stage algorithm was proposed. At the first stage, the simulated annealing algorithm is used with a limited number of iterations. At this stage, several work processes are executed in parallel. The initial solution of each work process is a part of the individuals of the current population. The results of the calculations form a new current population, which is transferred to the second stage. At the second stage, the main process processes the current population using the genetic algorithm and distributes its individuals to the work processes as initial solutions.

The algorithm of each workflow contains five subsequent steps of simulated annealing:

Step 1. We consider the initial temperature to be T_0 and the value of the iteration counter to be 1.

Step 2. As the initial mapping X, we select the solution passed by the master process and calculate the value of the fitness function.

Step 3. At the next iteration, we randomly select one of the vertices (i) of graph G. We iterate over the remaining vertices of the graph in turn. We swap the selected vertex i and the vertices being iterated. We calculate the increase in the objective function $\Delta\eta(X)$ when swapping the selected vertex i and the next vertex j. If $\Delta\eta(X) < 0$, then the replacement is fixed, and the transition to a new iteration is performed. If $\Delta\eta(X) > 0$, then the replacement is fixed with probability $e^{-\Delta\eta T}$, where T is the current annealing temperature, and the transition to a new iteration is performed. If the fixation does not occur, a new, $(j+1)$-th, vertex of the program graph is selected, and its places are exchanged with the vertex i selected at the beginning of the iteration.

Step 4. Temperature decrease according to the $T = T_0/t$ law.

Step 5. After a certain number of iterations, or if the fitness function $\eta(X)$ is fixed at a stationary value, we complete the algorithm, otherwise we proceed to Step 3.

The main process collects the results of all work processes into a new population. This process performs selection operations on the population, rejects the worst individuals, and forms new individuals by crossing. The obtained individuals, together with the individuals selected as a result of selection, form a new population. The new population is distributed among the work processes, and the next step of the algorithm begins.

The algorithm stops if after a given number of steps there is no improvement in the fitness function.

To assess the effectiveness of the proposed method, a simulation model of an autonomous fragment of the fog layer of the ecosystem supporting the mobile high-density Internet of Things was used. The coverage area was 100 km^2. The fragment contains the following devices:

– 12 active intelligent gateways, the reachability radius of the data transmission channel is 3 km;

– 5 fog servers with the ability to guarantee information reception within a radius of 1 km via 10 independent channels;

– 150 fog mobile devices.

For one hour, the gateways are constantly receiving information received from MHDIoT sensors and transmitted by boundary layer devices. Gateways form information flows of tasks that are transmitted to mobile devices of the fog layer that are within reach. The trajectories of mobile devices are random variables.

To assess the quality of management of computing resources in the fog layer, the load on intelligent gateways was simulated.

Two control options were considered:

– option 1: standard control, in which the task is transferred to the nearest available server;

– option 2: control using a two-stage task distribution algorithm.

Different options for forming stable mobile clusters were also chosen. In the first option, the average number of free fog devices h was 20% of the total number, in the second – 50%.

The load balance and average task dwell time in the fog layer indicators were evaluated. The normalized area of the relative deviation from the balanced load ($\eta_{relative}$) was considered as an indicator of load balance. The second indicator was estimated by the ratio of the average task dwell time in the system for different control options ($WT_{average} = WT1_{average} / WT2_{average}$). In this indicator, the numerator corresponds to the proposed control method, and the denominator to the standard one. The simulation results for these indicators are shown in Fig. 3, 4.

When modeling load θ on intelligent gateways, it was formed in the amount of 0.1 to 0.9 fraction of the maximum possible load. The information flows of tasks formed by intelligent gateways were transmitted to foggy mobile devices discretely, with a step of 4 minutes.

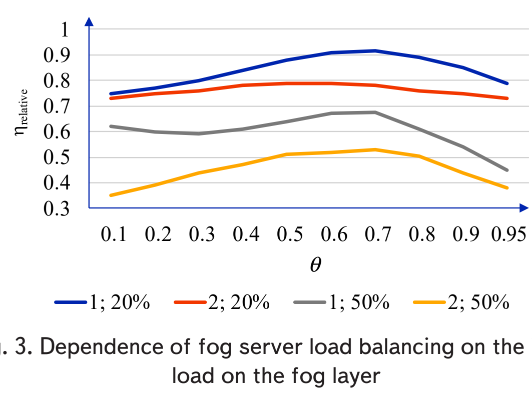

Fig. 3. Dependence of fog server load balancing on the input load on the fog layer

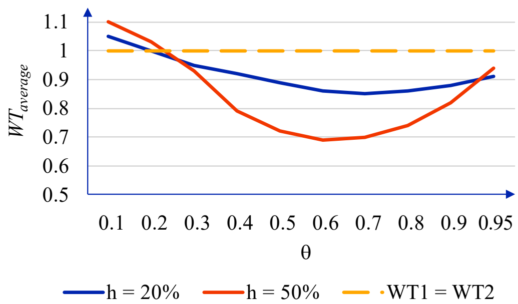

Fig. 4. Dependence of the ratio of the average task residence time in the system for different control options on the input load on the fog layer

6. Results of investigating the method for managing computing resources in the fog layer of the mobile high-density Internet of Things: discussion and summary

A mathematical model of the process of managing computing resources in the fog layer has been proposed. A feature of this model is its module hierarchical structure according to the basic levels of decision-making when managing computing resources (Fig. 2). The model is based on the architecture of the MHDIoT ecosystem (Fig. 1). The application of the principle of process decomposition by adjacent time intervals made it possible to formulate constraints (15) to (17) for the combinatorial optimization problem (18).

The criteria and quality indicators of the process of managing computing resources in the fog layer have been determined. A feature of the proposed quality indicators is the use of the concept of the area of the parameter of the time process (formulas (3), (4)). This made it possible to estimate the total load on nodes for a certain period of time and formulate the objective function (30) of the optimization combinatorial problem.

When devising a method for managing the computational resources in the fog layer of the mobile high-density Internet of Things, the simulated annealing algorithm and the genetic algorithm were used. The main difference is the joint use of algorithms when finding a solution. This allowed us to reduce the time for searching for the distribution of the computational resource of the fog layer. The method is performed in 2 stages. At the first stage, the simulated annealing algorithm finds several autonomous solutions. At the second stage, the genetic algorithm strengthens the resulting population, which helps increase the convergence rate.

The proposed method was evaluated by load balance indicators and average task residence time in the fog layer. The normalized area of the relative deviation from the balanced load was considered as an indicator of load balance. The assessment of the effectiveness of the proposed method (Fig. 3, 4) showed the following results:

1. With a small load on the fog layer ($\theta < 0.2$), the proposed method balances the load significantly better than the classical one (Fig. 3). However, the average task residence time in the system with the proposed method is somewhat higher than with the classical method (Fig. 4).

2. With a load on the fog layer from 0.2 to 0.7 fractions of the maximum possible, the proposed method also balanced the load better than the classical one (Fig. 3). Also, with an increase in the load, the average task residence time in the system was less than with the classical method.

3. With a large load on the fog layer ($\theta > 0.8$), the indicators under consideration almost did not differ.

The results of our study of the proposed method can be explained by the use of the simulated annealing algorithm in combination with the genetic algorithm. Unlike [14, 15], which propose resource allocation algorithms for the edge and fog layers of IoT, the proposed method takes into account the features of high-density IoT. In addition, the proposed method, unlike [16, 20], takes into account requests from fog devices that do not belong to stable fog clusters.

Unlike [17], in which the TOPSIS method is used, the proposed method finds a solution significantly faster. Also, unlike [18, 21], in which all fog devices are clustered, our method distinguishes between free and clustered fog nodes. Unlike [19], in which balancing occurs by switching to less congested channels, the proposed algorithm takes into account the mobility of fog devices.

Thus, the devised method for managing computing resources in the fog layer of the mobile high-density Internet of Things has made it possible to reduce the imbalance of fog server load. That has made it possible to meet QoS requirements even with the high density of mobile devices.

However, the proposed results should be applied under the following restrictions:

– high density of IoT mobile devices;

– the presence of stable mobile clusters, the center of which is the fog server.

In addition, the proposed method is advisable to use if the average load on fog servers is in the range from 20% to 70% of the maximum possible.

As a drawback of this study, it should be noted that the work does not analyze the impact of the location of fog servers on the quality of balancing. To eliminate this drawback, it is necessary to expand the simulation model of the autonomous fragment of the fog layer of the MHDIoT support ecosystem with an additional procedure. This procedure should implement different options for the location of fog servers in the territory covered by MHDIoT.

The following will advance our study in the future.

First, it is necessary to investigate the impact on load balancing of information flow servers with different priorities. Second, it is necessary to conduct research on the selection of the optimal number of intelligent gateways and fog servers for the MHDIoT ecosystem.

7. Conclusions

1. A mathematical model of the process of managing computing resources in the fog layer has been proposed. A feature of this model is its module hierarchical structure according to the basic levels of decision-making in managing computing resources. The model is based on the architecture of the ecosystem of the mobile high-density Internet of Things. When constructing the model, the principle of process decomposition by adjacent time intervals was used. Its application has made it possible to carry out local optimization of the process of managing computing resources at short time intervals. This mathematical model has made it possible to formulate constraints for the combinatorial optimization problem and requirements for its objective function.

2. The criterion and quality indicators of the process of managing computing resources in the fog layer have been determined. A feature of the proposed quality indicators is the use of the concept of the area of the parameter of the time process. This has made it possible to estimate the total load

on the nodes for a certain period of time and to formulate the objective function of the optimization combinatorial problem.

3. A two-stage algorithm for distributing tasks of free fog devices across fog layer servers has been developed. The main difference of this algorithm is the combined use of the simulated annealing algorithm and the genetic algorithm. At the first stage, the simulated annealing algorithm finds several autonomous solutions. At the second stage, the genetic algorithm strengthens the resulting population, which helps increase the convergence rate. That has made it possible to reduce the search time for distributing the computational resource of the fog layer. The optimization of the process is carried out according to the criterion of balancing the load of fog servers. The results of our study have made it possible to assess the effectiveness of the devised method. With a small load on the fog layer, up to 20% of the maximum possible, the proposed method balances the load significantly better than the classical one. However, the average time a task stays in the system with the proposed method is somewhat higher than with the classical method. With a load on the fog layer from 20% to 70% of the maximum possible, our method also balanced the load better than the classical one. But the average time the task spent in the system was already 1.1–1.6 times less than with the classical method. With a high load on the fog layer, more than 80% of the maximum possible, the indicators under consideration were almost the same.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Authors' contributions

Heorhii Kuchuk: Conceptualization, Formal analysis, Supervision. **Oleksandr Mozhaiev**: Methodology, Validation. **Serhii Tiulieniev**: Supervision. **Mykhailo Mozhaiev**: Validation, Data Curation. **Nina Kuchuk**: Software, Writing – original draft. **Pavlo Khorobrykh**: Project administration. **Yurii Gnusov**: Writing – review & editing. **Yurii Horelov**: Visualization. **Vitalii Svitlychnyi**: Resources. **Oleksandr Bilyk**: Investigation.

References

1. Kaur, G., Balyan, V., Gupta, S. H. (2025). Nature inspired optimization of IoT network for delay resistant and energy efficient applications. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-95138-z
2. Kuchuk, H., Malokhvii, E. (2024). Integration of IoT with cloud, fog, and edge computing: a review. Advanced Information Systems, 8 (2), 65–78. https://doi.org/10.20998/2522-9052.2024.2.08
3. Zhurylo, O., Liashenko, O. (2024). Architecture and iot security systems based on fog computing. Innovative technologies and scientific solutions for industries, 1 (27), 54–66. https://doi.org/10.30837/itssi.2024.27.054
4. Kuchuk, H., Chumachenko, I., Marchenko, N., Kuchuk, N., Lysytsia, D. (2025). Method for calculating the number of IoT sensors in environmental monitoring systems. Advanced Information Systems, 9 (3), 66–72. https://doi.org/10.20998/2522-9052.2025.3.08
5. Bhajantri, L. B., Gangadharaiah, S. (2022). Heuristic-Based Resource Allocation for Internet of Things in Gateway Centric Multi-layer Fog Computing. ICT Systems and Sustainability, 567–579. https://doi.org/10.1007/978-981-19-5221-0_54
6. Liu, Q., Mo, R., Xu, X., Ma, X. (2020). Multi-objective resource allocation in mobile edge computing using PAES for Internet of Things. Wireless Networks, 30 (5), 3533–3545. https://doi.org/10.1007/s11276-020-02409-w
7. Yu, J., Hou, K., Zhang, H., Kostic, B., Yang, M., Nazif, H. (2025). A new energy-aware resources scheduling method for mobile internet of things using a hybrid optimisation algorithm. International Journal of Mobile Communications, 25 (2), 176–207. https://doi.org/10.1504/ijmc.2025.144192
8. Kuchuk, N., Kashkevich, S., Radchenko, V., Andrusenko, Y., Kuchuk, H. (2024). Applying edge computing in the execution IoT operative transactions. Advanced Information Systems, 8 (4), 49–59. https://doi.org/10.20998/2522-9052.2024.4.07
9. Muñoz, L. A., Berná Martínez, J. V., Asensi, C. C., Pastor, D. S. (2024). RESEARCH NOTES: Design of a Distributed and Highly Scalable Fog Architecture for Heterogeneous IoT Infrastructures. International Journal of Software Engineering and Knowledge Engineering, 35 (02), 195–215. https://doi.org/10.1142/s0218194025430016
10. Lea, R., Adame, T., Berne, A., Azaiez, S. (2025). The Internet of Things, Fog, and Cloud Continuum: Integration Challenges and Opportunities for Smart Cities. Future Internet, 17 (7), 281. https://doi.org/10.3390/fi17070281
11. Zhurylo, O., Liashenko, O., Avetisova, K. (2023). Hardware security overview of fog computing end devices in the internet of things. Innovative Technologies and Scientific Solutions for Industries, 1 (23), 57–71. https://doi.org/10.30837/itssi.2023.23.057
12. Kuchuk, H., Kalinin, Y., Dotsenko, N., Chumachenko, I., Pakhomov, Y. (2024). Decomposition of integrated high-density IoT data flow. Advanced Information Systems, 8 (3), 77–84. https://doi.org/10.20998/2522-9052.2024.3.09

13. Rezanov, B., Kuchuk, H. (2023). Model of elemental data flow distribution in the internet of things supporting fog platform. Innovative technologies and scientific solutions for industries, 3 (25), 88–97. https://doi.org/10.30837/itssi.2023.25.088
14. Kuchuk, H., Mozhaiev, O., Kuchuk, N., Tiulieniev, S., Mozhaiev, M., Gnusov, Y. et al. (2024). Devising a method for the virtual clustering of the Internet of Things edge environment. Eastern-European Journal of Enterprise Technologies, 1 (9 (127)), 60–71. https://doi.org/10.15587/1729-4061.2024.298431
15. Lee, B. M. (2025). Efficient Resource Management for Massive MIMO in High-Density Massive IoT Networks. IEEE Transactions on Mobile Computing, 24 (3), 1963–1980. https://doi.org/10.1109/tmc.2024.3486712
16. Kuchuk, H., Mozhaiev, O., Tiulieniev, S., Mozhaiev, M., Kuchuk, N., Lubentsov, A. et al. (2025). Devising a method for energy-efficient control over a data transmission process across the mobile high-density internet of things. Eastern-European Journal of Enterprise Technologies, 4 (4 (136)), 46–57. https://doi.org/10.15587/1729-4061.2025.336111
17. Sandanasamy, A., Charles, P. J. (2025). Dynamic load balancing through TOPSIS based optimal server selection and resource allocation in SDN IoT network. OPSEARCH. https://doi.org/10.1007/s12597-025-00985-z
18. Shamsa, Z., Rezaee, A., Adabi, S., Rahimabadi, A. M., Rahmani, A. M. (2024). A distributed load balancing method for IoT/Fog/Cloud environments with volatile resource support. Cluster Computing, 27 (4), 4281–4320. https://doi.org/10.1007/s10586-024-04403-9
19. Shuaib, M., Bhatia, S., Alam, S., Masih, R. K., Alqahtani, N., Basheer, S., Alam, M. S. (2023). An Optimized, Dynamic, and Efficient Load-Balancing Framework for Resource Management in the Internet of Things (IoT) Environment. Electronics, 12 (5), 1104. https://doi.org/10.3390/electronics12051104
20. Kuchuk, H., Husieva, Y., Novoselov, S., Lysytsia, D., Krykhovetskyi, H. (2025). Load balancing of the layers IoT fog-cloud support network. Advanced Information Systems, 9 (1), 91–98. https://doi.org/10.20998/2522-9052.2025.1.11
21. Kanbar, A. B., Faraj, K. (2022). Region aware dynamic task scheduling and resource virtualization for load balancing in IoT–fog multi-cloud environment. Future Generation Computer Systems, 137, 70–86. https://doi.org/10.1016/j.future.2022.06.005
22. Chiang, W.-K., Chih, Y.-H. (2025). A pareto-optimal heuristic algorithm to refactor IMS core for cloud-native NFV. Cluster Computing, 28 (8). https://doi.org/10.1007/s10586-025-05177-4
23. Ritter, F. (2023). Technical note: A procedure to clean, decompose, and aggregate time series. Hydrology and Earth System Sciences, 27 (2), 349–361. https://doi.org/10.5194/hess-27-349-2023
24. Hu, C., Sun, Z., Li, C., Zhang, Y., Xing, C. (2023). Survey of Time Series Data Generation in IoT. Sensors, 23 (15), 6976. https://doi.org/10.3390/s23156976
25. Sajjad, F., Rashid, M., Zafar, A., Zafar, K., Fida, B., Arshad, A. et al. (2023). An efficient hybrid approach for optimization using simulated annealing and grasshopper algorithm for IoT applications. Discover Internet of Things, 3 (1). https://doi.org/10.1007/s43926-023-00036-3
26. Seyedi, B., Postolache, O. (2025). Securing IoT Communications via Anomaly Traffic Detection: Synergy of Genetic Algorithm and Ensemble Method. Sensors, 25 (13), 4098. https://doi.org/10.3390/s25134098
27. Gupta, N., Sharma, V. (2023). Context Aware Hybrid Network Architecture for Iot with Machine Learning Based Intelligent Gateway. SN Computer Science, 4 (3). https://doi.org/10.1007/s42979-023-01736-x