Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 10, No. 2, April, 2026

Trusted AI-Based Method for Predicting Controller Load and PSO-Based Structure for Reducing Latencies

Vladimir Zh. Kuklin [1*], Islam Alexandrov [2], Maxim Mikhailov [1], Naur Z. Ivanov [2], Elena Yu. Linskaya [3]

[1] *Institute of Design and Technology Informatics, Russian Academy of Sciences, Moscow 127994, Russian Federation.*

[2] *NRC "Kurchatov Institute" - SRISA, Moscow 117218, Russian Federation.*

[3] *ISP RAS Research Center for Trusted Artificial Intelligence, ISP RAS, Moscow 127994, Russian Federation.*

Abstract

The objective of this study is to develop a trusted AI-based framework for predicting software-defined networking (SDN) controller load and optimizing fog/edge microservice orchestration to reduce end-to-end latency in dense 5G scenarios. The proposed approach integrates user-aware spatial clustering with evolutionary resource selection to maintain stable quality of service (QoS) under high mobility and traffic variability. In the analysis stage, k-means clustering partitions users into spatial sectors and identifies sector centroids. Particle swarm optimization (PSO) is then applied to fog-node selection, resource sizing, and adaptive microservice placement and migration. To enhance system resilience, a recurrent neural network (RNN) is employed to forecast SDN controller load using correlation-informed features extracted from service-channel dynamics. Numerical experiments on heterogeneous fog-node topologies indicate that the framework reduces microservice execution time by 69% relative to baseline placement strategies under identical load conditions, while controller-load prediction attains an RMSE of 0.00387. These findings confirm the effectiveness of both the latency-reduction mechanism and the controller-load estimation workflow. The novelty of this work lies in the unified optimization of microservice placement, migration, and SDN controller-load anticipation within a single reproducible architecture, extending existing fog and edge orchestration approaches that typically address these components as independent subproblems.

Keywords:

Ultra-Low Latency;
Fog Computing;
Particle Swarm Optimization;
SDN Controller;
Load Forecasting;
Software-Defined Networking.

Article History:

Received:	02	December	2025
Revised:	19	February	2026
Accepted:	26	February	2026
Published:	01	April	2026

1- Introduction

Fifth-generation (5G) and future 6G networks must support ultra-reliable low-latency communications (URLLC) with end-to-end delays of 1-4 ms. Conventional centralized cloud architectures struggle to satisfy these constraints due to propagation delay, backhaul/core-network congestion, and limited responsiveness under highly dynamic traffic. Recent surveys demonstrate that Multi-access Edge Computing (MEC) and fog computing paradigms address this challenge by positioning computational resources closer to end users, thereby reducing latency from tens of milliseconds to the sub-5ms range required for tactile Internet applications [1].

However, achieving optimal service placement in fog computing environments remains a fundamental challenge. Existing approaches predominantly treat service migration as monolithic processes without addressing component-level microservice migration dynamics. Prior work suggests that current methods fail to systematically integrate user clustering with fog node selection optimization, resulting in suboptimal resource allocation in heterogeneous network topologies. Furthermore, service migration strategies proposed in recent literature lack predictive capabilities to anticipate load variations before performance degradation occurs [2].

* **CONTACT**: kuklin_vladimir_ran@mail.ru

DOI: https://doi.org/10.28991/ESJ-2026-010-02-017

Thus, in addition to describing the familiar operating principles of mobile networks, 5G/IMT-2020 networks combine various technological solutions and concepts. The use of multi-access edge computing (MEC) and fog computing architectures can ensure the placement of computing resources in proximity to users, thereby reducing signal propagation latencies [3]. However, most existing frameworks decouple service placement from traffic forecasting. Standard placement algorithms typically optimize for static network states, neglecting the continuous spatial shifts of user groups inherent in 5G. Additionally, while deep learning improves prediction accuracy, these forecasts are rarely used to initiate service migration before network congestion occurs. As a result, there is a lack of integrated solutions that combine user clustering with predictive load management to maintain low latency [4].

The increased interest in neural networks for data transmission networks is driven by the desire to overcome the difficulties and limitations of 5G networks by predicting the load to anticipate latencies. Neural networks are increasingly adopted to address the limitations of traditional mathematical models, which often fail to capture the high heterogeneity of 5G traffic. Deep learning enables accurate load prediction and latency anticipation by learning complex, non-linear patterns directly from dynamic network environments [5].

This study uses three primary levers to reduce latencies: spatial demand localization to optimize the service placement topology, border node selection to ensure sufficient server accessibility, and behavior prediction of microservice migration before violating performance requirements. We propose a range of comprehensive solutions to the above tasks based on an integrated concept for distributed cloud computing. The K-means algorithm performs spatial clustering of users, thereby forming groups with common centers. Next, the minimum number of fog nodes with microservices is determined using the swarm optimization method, thereby minimizing latency. In the next stage, a recurrent neural network (RNN) predicts the user load dynamics for timely migration. The effectiveness of the developed approach was evaluated on a fixed topology using MSE prediction accuracy and the migration frequency method.

2- Literature Review

5G mobile communications and IoT are fundamentally new technologies, the creation and implementation of which have led to a significant increase in information exchange and, accordingly, an increase in the number of machines connected to network communication systems. Therefore, the architecture of the equipment and computing equipment must comply with the new requirements. Industry analysts Zhu et al. (2025) [6] have estimated that within a few years, the number of network devices will increase, and the volume of information generated will be measured in petabytes. Most classic centralized cloud systems operate with numerous strict limitations, particularly affecting applications that must be processed online. Therefore, latencies and line congestion prevent the availability of ultra-low latencies corresponding to 1-4 ms. Accordingly, it is necessary to distribute cloud computing and bring computing equipment closer to the end user and source of information.

In recent years, edge and fog computing have become more popular, offering unique solutions to the problem of significant latency in traditional cloud systems. In the Rejiba et al. (2019) [7] study detailed analysis of edge and fog computing for service migration is provided, which is mainly driven by the high mobility of specialized equipment. The researchers emphasized the need to bring computing equipment closer to the source of information. This can support applications that are sensitive to latency, such as autonomous vehicles. Simultaneously, this requires a solution to a key problem: almost all existing methods treat service migration as a monolithic process without considering the specifics of the component migration of a defined service.

Most available options for integrating computing usually face significant limitations, primarily because of the architecture and functionality of the processes. Therefore, a variation of a specific model that can function in networks with relatively low energy consumption is proposed in this study. In this case, the entire set of edge servers provides computational support for IoT equipment, which has limited resources. However, this option is relatively static and cannot organize the flexible migration of services above a rapidly changing network configuration caused by high client mobility and unstable network equipment loads. In addition, studies by Bellavista et al. (2019) [8] and Mahmoud et al. (2021) [9] show, it is difficult to predict the location of equipment responsible for performing special calculations based on the behavior of average customers.

Service migration is a complex and intractable problem in edge computing. Solving this requires a unique balance, which involves maintaining the stable operation of each application, reducing potential downtime, and making highly efficient use of available resources. Modern reviews provide a good description of the entire range of identified limitations of the current study. Industry experts have pointed out that many solutions focus on the complete migration of unified services or the active relocation of a conditional application, considering that the client is constantly on the move. However, in such situations, it is difficult to use combined approaches and consider the specifics of moving individual services. Toumi et al. (2021) [10] have also emphasized that migrating services is an intractable problem, especially when high-quality services, reduced application downtime, and continuous interaction in a microservices environment are required. Without component-based and adaptive migration, maintaining the required latency is difficult.

In general, migration is determined by multiple interdependent factors, such as computing equipment rental, information container duration, considering the volume and operating mode of the application, optimal use of the resource base among competing ap-plications, and others. However, Pallewatta et al. (2023) [11] have not developed a systematic solution to optimize migration processes and consider system parameters, given that the network topology and load are highly heterogeneous. In addition, there are difficulties with the preliminary calculation of network equipment computing capabilities because it is difficult to predict its capabilities at the end of the migration processes.

Many studies classify approaches to equipment distribution based on optimization (numerous precise methods, simplification techniques, and machine learning benefits) and target metrics (various latencies, self-cost, energy consumption, reliability, etc.). However, the researchers Taleb et al. (2025) [12] and Jasim et al. (2024) [13] said even the methods listed above cannot optimally distribute the resource base and accurately determine user clusters.

Owing to the expansion of MEC standards by the European Telecommunications Standards Institute (ETSI), it is usually possible to support the migration processes of a conditional application and ensure its stable operation. Therefore, container technologies are the most promising for directly migrating most existing services. This direction is achieved mainly through the MEC-based architectural extension, which contains a series of new components that adjust the lifecycle of the information container and manage migration in the edge-node environment. However, this approach does not offer a single solution for systematically determining the computing capabilities. This approach does not allow for the pre-clustering of client devices, as was implied by Barbarulo et al. (2022) [14] and Escolar et al. (2021) [15] which can identify key areas of service distribution.

A programmatically configurable network and simplified network functionality are components of fifth-generation telecommunications systems. One of the most popular approaches to computer network management as shown in Farooq et al. (2023) [16] and Kerimkhulle et al. (2023) [17] is the use of a software-defined networking (SDN) structure. The main structure of this methodology is the separation of data management structures into different planes, with each plane being centralized to maintain the network state. Network function virtualization (NFV) technology provides the necessary flexibility to achieve this. The combined use of these methods optimizes network performance. Load balancing within a single layer helps distribute the load, ensuring a more stable network performance, which is particularly essential for networks with ultra-low latency.

Specialized literature describes numerous methodologies capable of optimizing, to varying degrees, the resource base of a deployed network with a specific configuration. Machine learning is used for this purpose, helping to predict probable changes in the network load and correct virtual functions, allowing clustering, forecasting, and adjusting for 5G network traffic. For this purpose, various works like Troia et al. (2019) [18] and Le et al. (2018) [19] on neural network architectures (primarily recurrent and convolutional networks) have been used to analyze time series of information exchange. However, at present, only a few studies have sufficiently examined the relationship between service flow and controller load, making it difficult to predict and model network behavior.

To plan the projected load on the SDN controller, including CPU (central processing unit), RAM (random access memory) parameters, it is necessary to predict performance based on various latencies and bandwidth determined by the network configuration and anticipated patterns of Internet traffic. However, Jiang et al. (2024) [20] showed that the accuracy of these forecasts (average error of 7–13%) remains low, especially considering ultra-low latency, where deviations of 1-2 ms disrupt the normal functioning of the Service Level Agreement (SLA) and significantly worsen network service. Notably, the literature does not offer universal, efficient tools for predicting the SDN controller load from service counters and their use in orchestration. The Aouedi et al. (2025) [21] work also does not provide predictions based on an in-depth analysis of operational flow metadata, considering the relationship between the load and system parameters.

In the tasks of resource allocation and planning for edge and fog computing, optimization methods minimize data transfer latencies and stabilize the load on computing nodes. Studies like Wu et al. (2025) [22] have shown that the use of particle swarm optimization (PSO) algorithms can reduce the average latency by 20-25% compared to similar practical techniques under static loads. The PSO algorithm can optimize resource-based distribution processes, plan jobs, and balance the loads. However, this toolkit does not involve pre-grouping users, which reduces its potential, because this approach can reduce the overall migration latency.

K-means was selected for spatial user clustering due to its computational efficiency and convergence guarantees, which are critical for real-time decision-making in 5G networks with monitoring frequency of 1 Hz. Unlike density-based methods (DBSCAN), Wu et al. (2025) [23] and Sousa et al. (2025) [24] showed that k-means produces compact, spherical clusters that naturally align with radio cell coverage patterns in MEC/fog deployments, where fog nodes serve geographically localized user groups. The assumption of spherical clusters is justified by the isotropic propagation characteristics of wireless signals in 3D indoor/outdoor environments. Alternative methods like DBSCAN, while capable of discovering arbitrary-shaped clusters, introduce computational overhead for distance calculations and require manual tuning of epsilon and minPts parameters, which is impractical for dynamic fog topologies with variable user density.

For SDN controller load prediction tasks, it is worth using a recurrent neural network (RNN) with the ability to disable latency-related features. The use of such technologies is an active area of development. Works like Aleisa et al. (2025) [25] and Akilandeswari et al. (2025) [26] have shown that, for example, flow-level characteristics such as packet size, interflow processing time, or application-level behavioral patterns can improve classification accuracy when working with high traffic-latency ratios. However, it is worth considering that the downside of using such characteristics is the requirement for deep packet inspection and maintaining the active flow state, which causes additional load in ultra-low-latency networks [25, 26]. Furthermore, recent works like Jiang (2025) [27] on traffic management shows that, for load prediction tasks, metrics for data and packets transmitted can effectively help ensure the connectivity of microservices, thereby ensuring the stable operation of a network with ultra-low latency.

Literature sources do not fully describe the capabilities and potential of cloud computing distribution because of incomplete integration with other tools. Many studies de-scribe the principles and features of edge and fog computing individually; therefore, it is not about combining the capabilities and advantages of MEC, NFV/SDN, and fog computing to account for the spatial location dynamics of the network equipment. The migration processes for existing services are covered in the same way: specialists have not yet developed a systematic solution to calculate the optimal nodes for migration efficiently and consider customer concentration to distribute the available fog equipment properly.

The proposed method addresses this problem by creating an integrated concept that enables communication between user clustering, node selection optimization, and load forecasting. For the optimal performance of these methods, a combination of K-means (necessary to determine customer concentration areas) and PSO optimization (for selecting the optimal fog equipment for current conditions) methods are used. This study proposes a recurrent neural network (RNN) to calculate the load on the SDN controller with sufficient accuracy, thereby increasing the reliability of communication networks.

3- Material and Methods

3-1- Methodology Ensuring Distributed Computing Communication and Service Support

To ensure synergy between technological solutions such as MEC, Fog, NFV, and SDN, several options for their simultaneous application have been proposed. The pro-posed system comprises blocks for collecting telemetry data (CPU and RAM ByteCount), a block responsible for decision-making (RNN + PSO) utilizing K-means clustering, and an orchestration block that manages migration and network flows, initiating commands for data collection, decision-making, and migration. Studying this problem from the perspective of classical hierarchies, such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), the advantages of network and computing infrastructure communication have been successfully implemented at the PaaS level. In addition, in most cases, the concepts listed above ensure the consistency of the proposed service [28].

Figure 1 shows a variation of the proposed concept that enables communication of service support and distributed computing. By combining network equipment, it is possible to distribute computing more efficiently to network edges and the end devices.

The proposed option provides an adjustment via an infrastructure interface using a computing equipment coordinator and an NFV/SDN network regulator. Using this option ensures the high quality of every service provided by the operator and network. The system interacts through the coordinator's API, which receives requests for microservice placement and returns fog nodes and migration routes. The NFV/SDN controller is responsible for creating these virtual functions. In addition, a certain degree of detachment from the physical resource base (for example, automatic formation of a service application) largely ensures the implementation of the system logic of operators that adjust the operation of controlled equipment [29, 30]; In other words, autonomous services process artificial intelligence (AI) category information.

Consequently, monitoring and correction are provided by an infrastructure application (software) that uses machine intelligence and automates controlled processes. In addition, networks can begin to respond independently to the current situation, ensuring load redistribution, when necessary, guided by quality-of-service (QoS) rules. The rules included a maximum delay of 4 ms, as well as monitoring of the CPU and RAM usage, ByteCount, and PacketCount at a frequency of 1 Hz. These networks consider the entire range of their performance indicators. Their monitoring can help generate the predictive analytics required to increase or reduce network capacity (resource base and calculations implemented by the entire cloud configuration infrastructure).

The structure of this concept (Figure 1) contains special components that successfully perform various calculations. The structural part of the MEC is built on a clearly ordered hierarchy of components, where lower-level computing cloud layers are subordinate to higher-level ones. In contrast, fog computing does not inherently possess such a hierarchy.

However, considering the technical computing potential of fog equipment, the order of its spatial distribution, and communication in a machine-to-machine environment, significant synergy can be achieved. This study highlights a "communication gateway" that aggregates and routes requests between user IoT devices, providing communication with the Internet, for which a low-level server system called "Micro Cloud/MEC" is used. The main functions of the gateway include receiving requests from user devices, translating data formats, and load balancing during periods of high traffic. Figure 1 shows the most likely communication option.

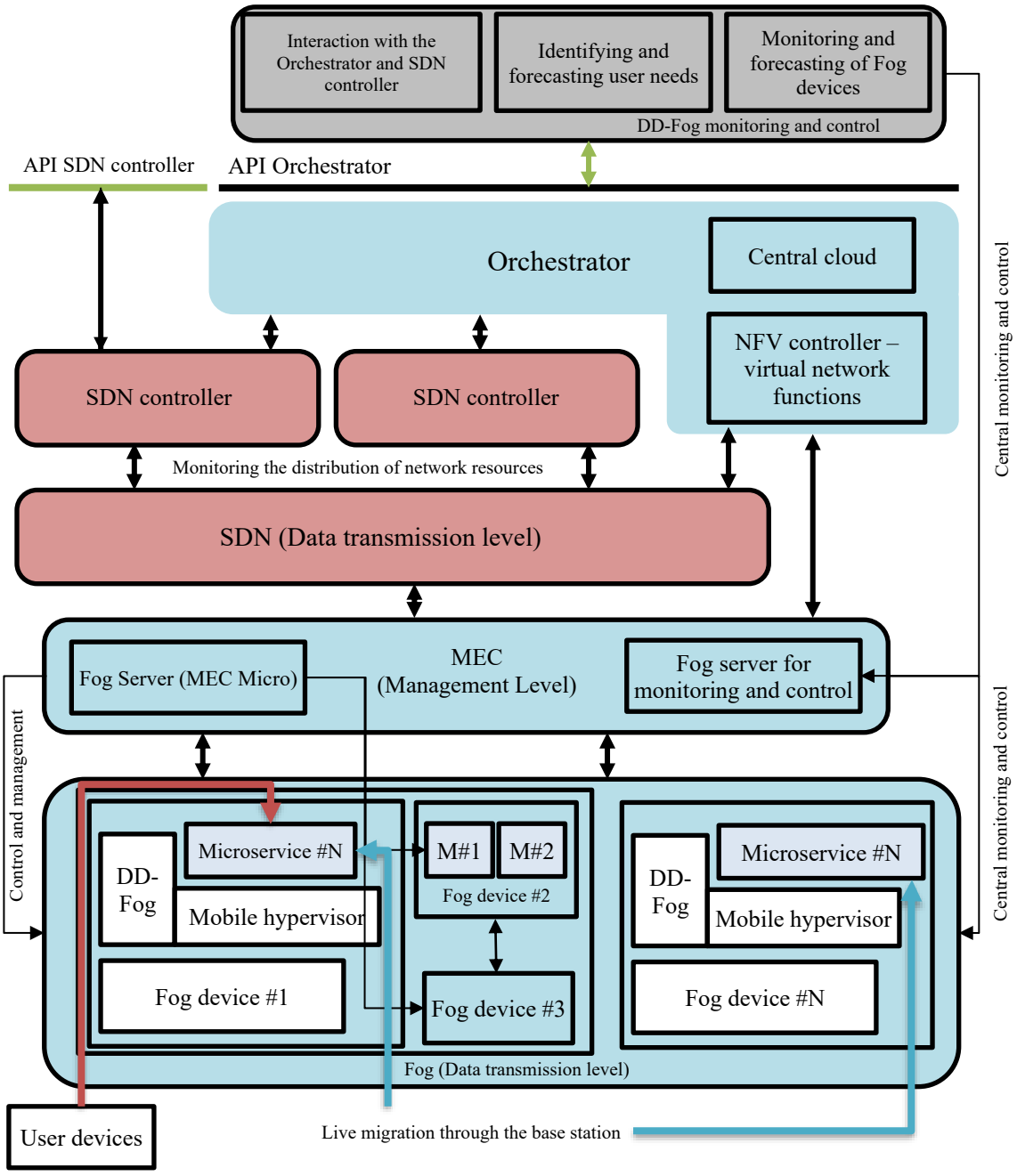


Figure 1. External view of the proposed concept, which fully describes the basic principles of communication

Note that the structure under consideration is capable of distributing fog equipment in a dynamic mode and, therefore, can adjust the resource intensity of all fog sectors in time. Simultaneously, the migration process of services offering a specific range of services was monitored. When detecting a new node, the system registers the node in the SDN controller, which manages the topology via a representative state transfer application programming interface (REST API). Fog nodes report their resources to the coordinator, which recalculates the position of the microservices. This concept ensures communication between structural components each time new fog equipment units are added. Figure 2 presents the communication features in the form of a schematic representation of the information exchange.

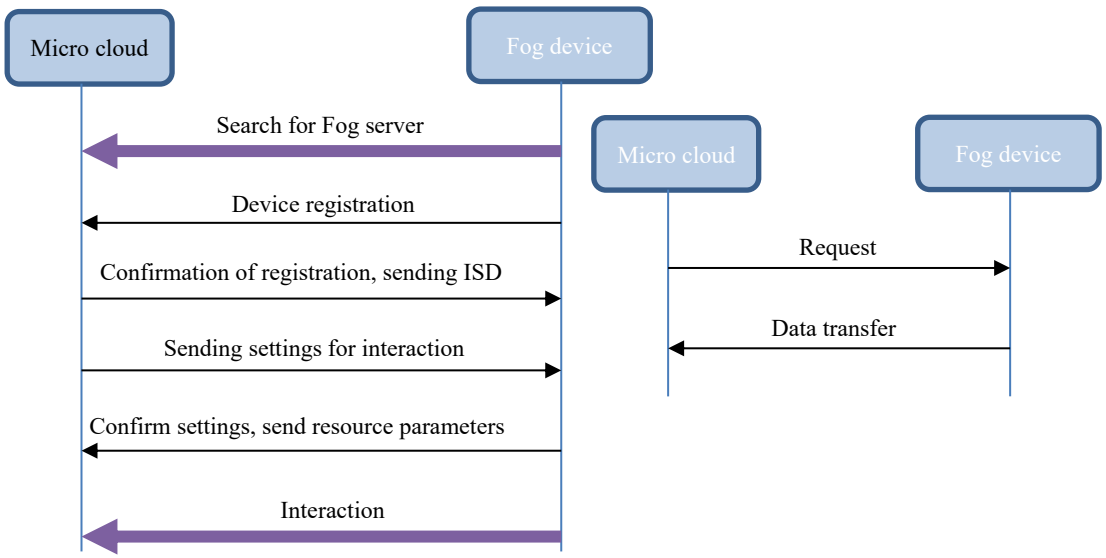


Figure 2. Schematic representation of the information exchange features in the environment of the structural components of the concept under consideration

This concept relies on the dynamic redistribution of fog equipment. The initiator is a fog device that generates a request in JavaScript Object Notation (JSON) format, which transmits CPU, RAM, and ByteCount values, wireless technology codes, and supported formats of microservices and orchestration systems. Simultaneously, the MEC structure generates a node identifier. The transition from the first fog sector to the second or the first addition of a device is accompanied by the fog equipment sending a broadcast request to the system to identify the nearest MEC structure. Once identified, the fog equipment provides comprehensive information necessary to register the device in the fog sector.

Subsequently, the principal server machine of the fog sector considers its characteristics and assesses the probability of adding the given fog equipment. If the server machine decides to refuse due to insufficient resources, a new attempt can be made in 5 min. The fog server generates a corresponding response in the form of a unique key that approves the addition of fog equipment. The message generated in this way by the fog servers simultaneously forms the unique code required for further communication between the fog equipment and the predominantly static cloud components of the MEC, including the system that corrects service migration to uniquely identify all available fog equipment units in fog sectors if the physical relocation of any units is planned.

Next, the formed signal arrives at the fog servers, which configure the communication parameters, and the fog equipment sends an internal code confirming readiness. Subsequently, the fog servers are granted the right to use the computing resource base of the corresponding fog equipment and open access to other network resources if necessary. The system components then communicate with each other at higher levels. These components regularly transmit information packets to the MEC cloud regarding the fog sectors under their control and respond to requests sent by the MEC cloud. Here, the regularity of the requested data may vary and depend on the application of a specific type of fog sector information processing to track and adjust the available computing base. Thus, all mini-clouds that are parts of the MEC can receive data regarding the controlled fog sectors at regular intervals of 1 s, ensuring optimal load monitoring.

To identify sectors where customer requests for a particular service are concentrated and determine the available computing capacity, it is necessary to apply a list of specific methodologies capable of processing information exchange (Figure 3).

K-means clustering can be applied to identify such a sector. This process will be implemented in stages as follows:

1. First, a package of input information is formed: a specific position with the coordinates x, y, and z of the sectors. In parallel, sectors are established (including their required number). Next, clients (client equipment) are assigned to the nearest cluster center. For this purpose, a distance metric is used that spatially separates the sectors: $\sqrt{(x-xk)^2+(y-yk)^2+(z-zk)^2}$. Here x, y, z are the current position with coordinates of a specific sector.

2. The mass centers are determined. For this purpose, $C_{mx}=\frac{\sum_{i=0}^{l}x_i}{l}$, $C_{my}=\frac{\sum_{i=0}^{l}y_i}{l}$, $C_{mz}=\frac{\sum_{i=0}^{l}z_i}{l}$, belonging to a specific sector, is used, where l is the number of fog devices in the sector.

3. A comparison is made between the central mass areas and the predicted central areas of each cluster.
4. If the sectors compared in point number 3 match, the central areas of each sector are successfully identified, so each customer assigned to it is marked and becomes one of the cluster components. If the compared sectors do not match, points number 2, 3, and 4 are repeated cyclically until the desired result is achieved.

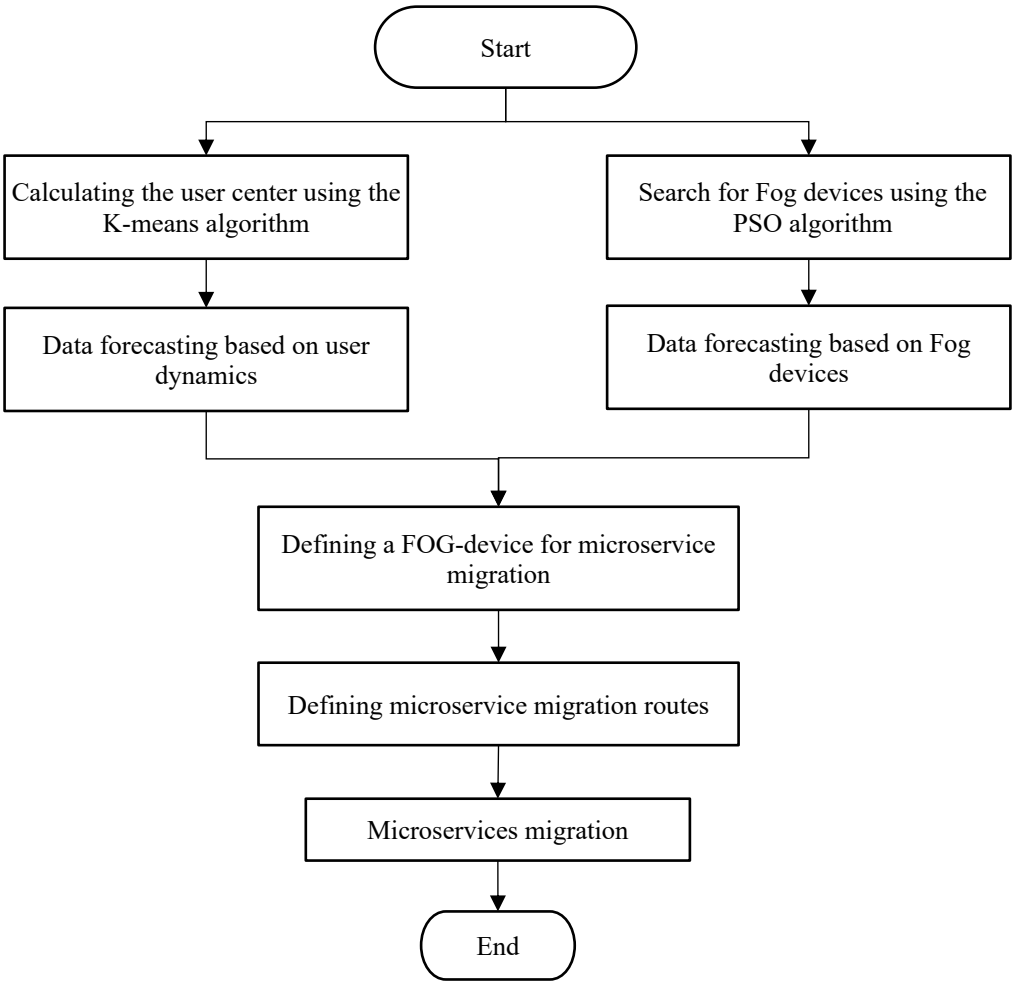


Figure 3. Schematic representation of the algorithm of the concept being considered

3-2-Specifics of Determining fog Computing Equipment to Ensure Live Migration of Services

To identify fog devices with available computing capacity and ensure appropriate migration of services, it is necessary to find the maximum optimization function that describes the state of fog devices. It is also logical to employ this methodology to deploy them over time and make them a part of a unified structure:

All individual components i are composed of several vectors: the current location in the D-dimensional $\overline{x_i} = x_{i1,x_{i1}}, \ldots, x_{i1}$, the best of the identified positions $\overline{p_i} = (\overline{p_{i1}}, \overline{p_{i2}}, \ldots, \overline{p_{iD}})$, and the directed velocity $\overline{v_i} = (v_{i1}, v_{i2}, \ldots, v_{iD})$. At the start of the method, the components are simultaneously and randomly initialized throughout the search area across the entire search space (the component velocities are also activated randomly). The created components move in the search area, described by simple relationships described in Equations 1 and 2. For each cluster, the algorithm solves the problem subject to CPU, RAM, and SLA constraints. The function is stopped when all steps t and all particles i have been processed. The method ensures regular updating of the entire swarm at all time intervals:

$$v_{id} = v_{id} + c\varepsilon_1(p_{id} - x_{id}) + c\varepsilon_2(p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where c is a value expressing constant acceleration, ε_1 and ε_2 are random numbers in the range [0, 1], p_{id} is the best position of all those passed by the components, p_{gd} is the location determined by the neighboring component within the neighborhood. The update is briefly described by the method shown in Figure 4.

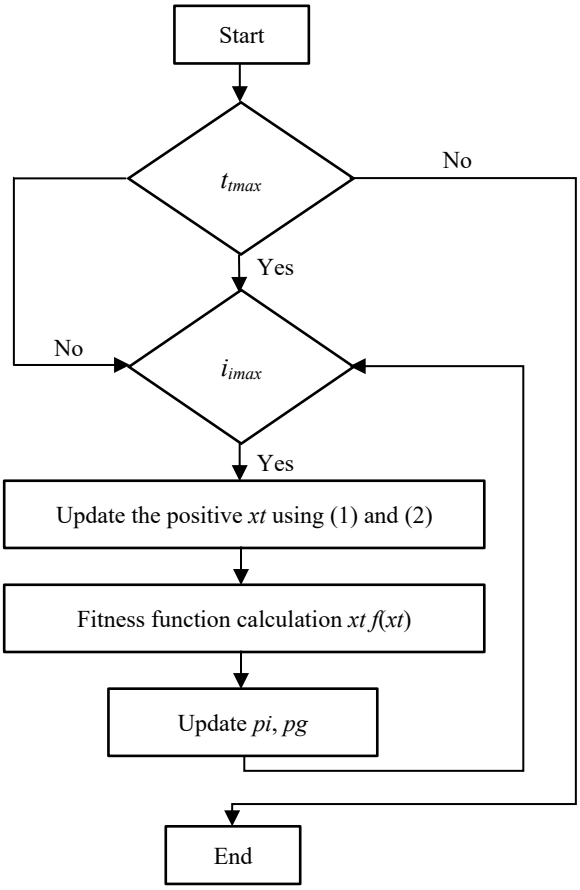


Figure 4. The procedure for updating the methodology required to optimize the swarm

The update is incremental in nature, commencing with the recalculation of speed and subsequently progressing to the update of position. The velocity of the particles is fixed at the maximum permissible value, thus circumventing any potential issues that might arise from the absence of congruence between the algorithm and the parameters under consideration. The values delineated in Equations 1 and 2 are likely to result in values that approach infinity.

Consequently, it is necessary to identify a basic list of characteristics that describe all tracked fog equipment units. At the global level, this list corresponds to the duration of service implementation by selecting a specific fog sector within which the required service must migrate. Therefore,

$$T = \sum_{i=1}^{n} W_i * TS_i \quad (3)$$

where T is the calculated characteristic, ms, W_i is the weight of the characteristic being sought, TS_i is the characteristic expressing the operating mode of the fog device unit, ms, n is the number of characteristics being sought, TS_1 is the propagation latency, ms, and TS_2 is the service request processing time, ms.

To ensure the robustness of the proposed fog node selection mechanism, the Particle Swarm Optimization (PSO) parameters were selected based on the canonical convergence analysis by Clerc and Kennedy, adapted for the low-latency constraints of 5G environments. The velocity of particles is clamped to a maximum value set to 20% of the dynamic range of the variable on each dimension. This constraint prevents 'swarm explosion,' where particles oscillate chaotically without settling, thereby ensuring the algorithm's stability.

The utilization of weighted sums is substantiated by the fact that this approach compensates for latency discrepancies, a prerequisite for effective operation in dynamic fog networks. Furthermore, the implementation of stringent SLA requirements has the potential to impose restrictions on solutions during periods of peak loads, which is an unacceptable scenario. Consequently, the proposed method of weighted sums facilitates the optimal selection of a solution.

Each indicator exerts an equivalent influence on the analyzed characteristic. Consequently, the weights of the characteristics (W_i) correspond to 1/2, and the summed weights are constrained to a maximum of 1 (one) and remain constant. A Pareto sensitivity analysis was conducted, which demonstrated that the configuration with equal weights provides the optimal compromise between propagation delay and processing costs. Whilst the implementation of extreme weights does result in a minor reduction in delay, it must be noted that the associated resource costs are disproportionately high. In the calculation process, options that are not suitable in terms of CPU, RAM, and bandwidth are disregarded. The purpose of this optimization characteristic T is to find the minimum values through the PSO algorithm. The desired characteristics are calculated in ms, and therefore:

$$T = \sum_{i=1}^{2} W_i * TS_i = 0.5TS_1 + 0.5TS_2, \quad (4)$$

In this instance, the first characteristic is identified through the utilization of a time tracker and the proposed concept, while the second characteristic is identified by network equipment and corresponds to the duration of customer request processing. The function does not involve the response delay in the control plane, because PSO decisions are made based on a load forecast.

3-2-1- Problem Formulation

Let $F = \{f_1, f_2, \dots f_M\}$ be the set of available fog nodes, and $S = \{s_1, s_2, \dots s_N\}$ be the set of microservices to be migrated. The objective is to find an optimal assignment matrix, where $x_{i,j} \in \{0,1\}$ is a binary decision variable:

$$x_{i,j} = \begin{cases} 1, & if\ \ s_j \text{ assigned to } f_i \\ 0, & \text{otherwise} \end{cases}$$

To implement the proposed methodology, a microservice placement problem is defined in the Equation 5. The optimization goal is to select a network configuration that satisfies strict latency requirements (SLA) and hardware capacity constraints (CPU/RAM), as shown in Equation 6. The mathematical formulation of the resource allocation problem is as follows:

$$\sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij}\ (0.5 * TS_1 + 0.5TS_2) \Rightarrow min\ \Phi \quad (5)$$

$$\begin{gathered} \sum_{j=1}^{N} x_{ij} * CPU_j^{req} \leq CPU_i^{total}, i \in \{1, \dots, M\} \\ \sum_{j=1}^{N} x_{ij} * M_j^{req} \leq M_i^{total}, i \in \{1, \dots, M\} \\ x_{ij} \leq T\{1, \dots, N\}_{max} \\ \sum_{i=1}^{M} x_{ij} = 1, j \in \{1, \dots, N\} \end{gathered} \quad (6)$$

3-3- Methodology for Forecasting the Workload of Controllers in Software-Defined Networking

The network traffic forecasting module must operate in 5G under low-latency enhanced mobile broadband (eMBB) and ultra-reliable low-latency communications (URLLC) scenarios, with a maximum latency of up to 4 ms. The SDN controller collects data via REST API on ByteCount and PacketCount at regular intervals of 1 s.

To guarantee the requisite quality and reliability of each service, with a view to enhancing the accuracy of network load forecasting, considering the characteristics of ultra-low latency networks, it is imperative to achieve stability in adjustments to the 5G network and its new variations.

This study employs correlation analysis as a method for verifying the relationship between metadata changes and the load on the SDN controller. The matrix *X*, for which weighted estimates are calculated using Equation 8, is represented in Equation 7.

$$X = \begin{array}{cccc} \text{ByteCount} & \text{PacketCount} & \text{CPU} & \text{RAM} \\ x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{i1} & x_{i2} & x_{i3} & x_{i4} \end{array} \quad (7)$$

$$u_{ij} = \frac{(x_{ij} - \mu_1(x_j))}{\delta(x_j)}, \quad (8)$$

where δ is the dispersion; μ is the mathematical expectation; *j* is the observation parameter represented in the first line of Equation 6, and *i* is the observation number.

The correlation moment and coefficient parameters, as represented in Equations 9 and 10, respectively, are utilized to calculate the correlation between random variables. The existence of a functional relationship between two parameters is indicated by the correlation coefficient being in the range [0, 1]. It is demonstrated that the higher the value, the stronger the correlation.

$$\xi_{jk} \frac{1}{n} \Sigma_{j=1}^{n} \left(x_{ij} - \mu_1(x_j)\right)\left(x_{ik} - \mu_1(x_k)\right), \quad (9)$$

$$\rho_{jk} \frac{1}{n} \Sigma_{j=1}^{n} (u_{ij} x_{ik}), \quad (10)$$

where *n* is the number of observations.

To test the load, let us use an analytical system with the OpenFlow protocol. SDN functions as the network controller, thereby achieving the abstraction of network functionality. In the first option, the principal functional constituent responsible for effecting the correction is a regulator that provides logical communication and tracks each protocol of communication. The information exchange level is chosen using MikroTik switches. The datasets were formed based on requests received through the API, followed by filtering by service flows to form analytical models, tracking ByteCount and PacketCount metrics. Figure 5 shows the final segment structure.

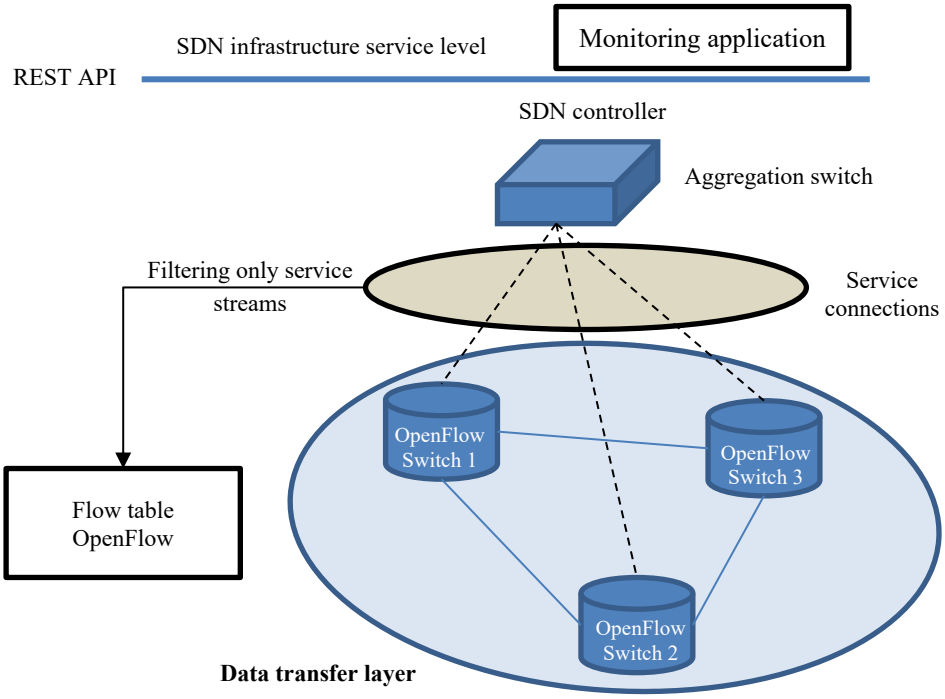


Figure 5. Final segment structure with representation of the components under study

To solve the current problem, it is necessary to use a neural network that provides forecasts of loads. Figure 6 shows a schematic representation of the created artificial neural network (ANN).

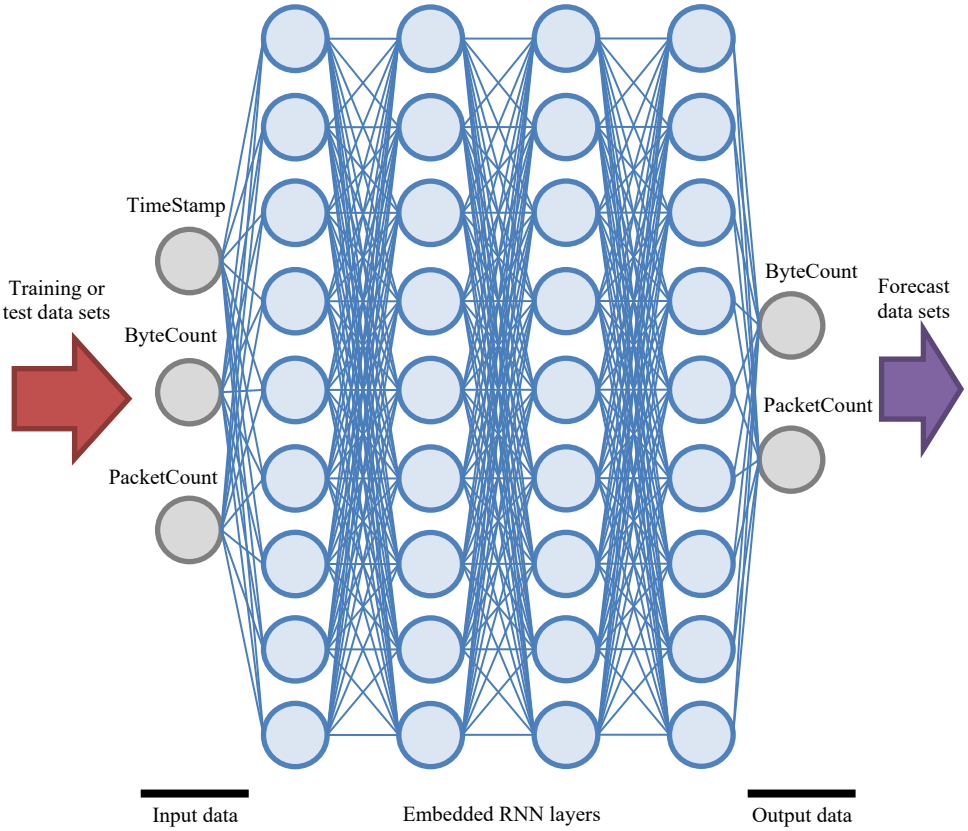


Figure 6. Schematic representation of the created ANN

The following considers the operation of the created ANN. The surface neural layer receives a packet of information from the developed dataset for machine learning (DataSet ML). The neurons of the surface layer communicate with the first layer of the neural net-work, using a fully connected architecture. The formation of predicted values is the function of a pair of neurons, and the architecture is characterized as recurrent. The data are prepared using a standardized assessment.

The input channel of the neural network perceives information that has a strictly fixed dimension. For this purpose, the initial information list is divided into several sections consisting of 200 input points. The initial list is divided into two separate streams without mixing: the first stream performs training, and the second stream performs testing. The test sample size is 20% of the training sample.

The training of the neural network involves the following parameters:

- The modernizer is Adam.
- The number of epochs is 20.
- The batch size is 300.
- The number of parameters per iteration is 1023.
- The learning rate is 0.0024.

4- Results

4-1- Creating a Methodology Model Responsible for Searching for Customer Concentration Centers and Identifying fog Equipment Units, Ensuring Subsequent Service Migration

To create a model of the proposed methodology, a simulation and auxiliary libraries were pre-formed at the software level. The K-means algorithm presented earlier was used. During the first stage, the simulation generated information about customer equipment.

The initial data packages became the basis for generating customer information:

1. The number of areas within which customer concentration is observed;
2. The predicted areas were distributed randomly according to the probability of assumptions.
3. Consequently, the formation of positions with the coordinates of the equipment located in each cluster was completed. Thus, the first sector covers ten units of customer equipment, the second – fifty; the third – fifteen; the fourth – thirty; the fifth – fifty units of customer equipment. Figure 7 schematically shows the entire set of clusters in 3D spatial coordinates.

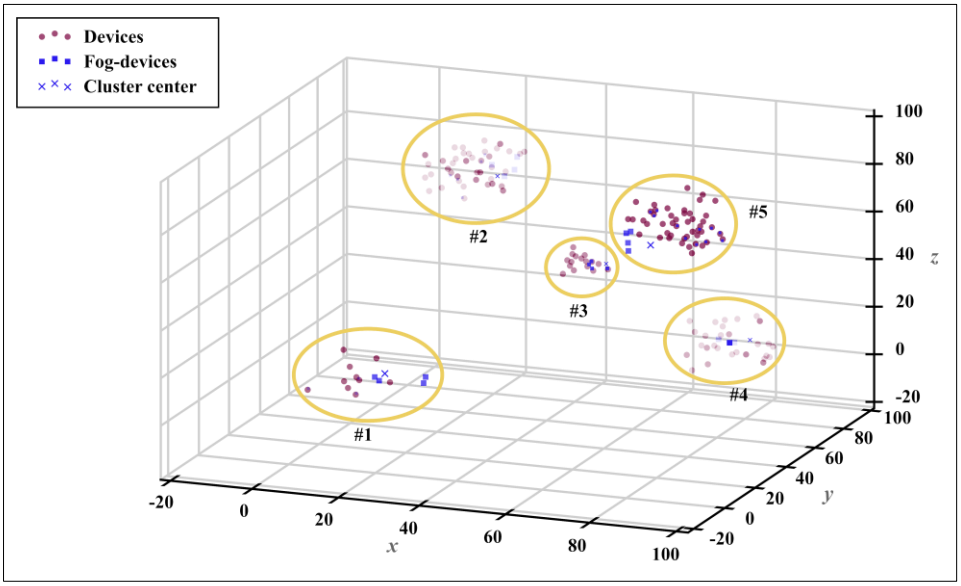


Figure 7. Information generated regarding customer equipment units (position 1)

The description of the generated information is of interest: in terms of practical modeling, clusters 1 and 4 cover the concentration of customers located in the cafeteria, mainly on the first floor of the building. Customers located on other floors, namely office premises, are also considered here. The remaining clusters also cover the

concentration of customers located above the office premises. This approach to cluster division directly influences the propagation delay because clusters 1 and 4, for example, provide a shorter distance to the fog nodes located at their level.

In creating the K-means model, as outlined in the theoretical foundations, it was necessary to determine a list of unpredictable sectors within which customer concentration can be detected. The algorithm clusters the x, y, and z coordinates obtained using fog nodes, thus enabling the determination of areas of user concentration regardless of movement. In the subsequent phase, the K-means methodology was developed (a necessity for identifying customer concentration), and the probable dispersion of customer equipment in relation to the current location of its owners was ascertained. This is due to the physical characteristics of wireless signal propagation in fog computing environments. In open or semi-open 5G network scenarios (e.g., campuses or shopping malls), the radio coverage area of a fog node is approximated by a spherical or convex region due to isotropic signal attenuation. K-means naturally partitions user space into cells that are consistent with such coverage patterns.

Figure 7 shows the central sectors of each cluster, which were calculated using the K-means method. Purple dots represent customer equipment. The probable central sectors of each cluster, which were identified randomly during the launch of the methodology under consideration, are marked with blue crosses. Green stars represent the central sectors. Information regarding the calculated radii of each cluster is expressed in a bar chart (Figure 8), the purpose of which is to determine the density of each cluster.

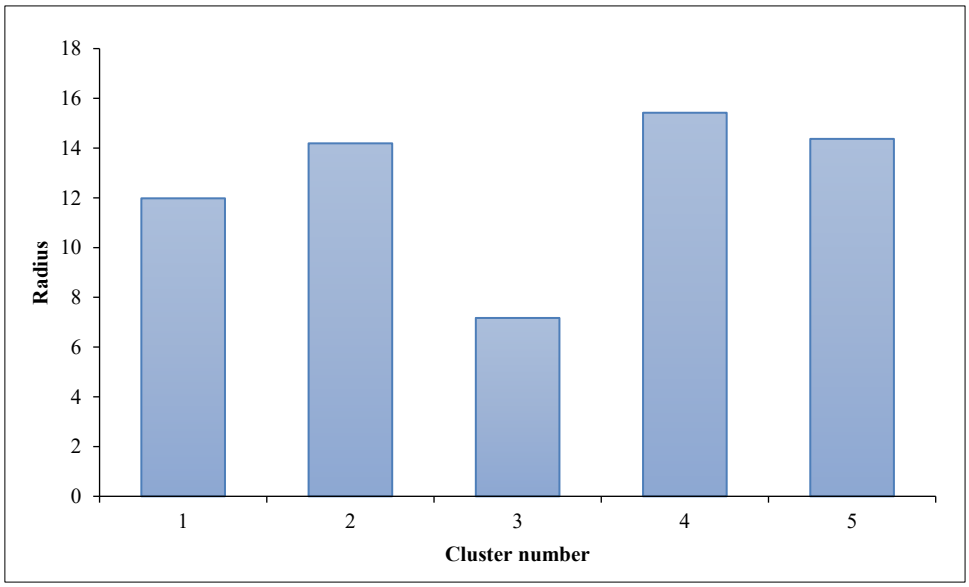


Figure 8. Calculated radii of each cluster

The results of the calculations confirmed that the third sector, covering fifteen units of customer equipment, has the maximum density, while the fourth sector covers three dozen units of customer equipment. Note that when analyzing the density of equipment distribution per square meter, the fifth sector demonstrates the maximum density. In addition, when comparing the distances separating customers, this sector is adjacent to the fourth, which is the largest.

To obtain data on the devices, values were generated using the previously specified PSO algorithm based on function (4). The fifth sector covers fifty units of client equipment with a list of characteristics formed in the ranges $TS_1 \in$ [0.5 to 10] ms and $TS_2 \in$ [0.2 to 2] ms.

The results of the PSO methodology show that the 13th unit of network fog equipment, with characteristics TS_1 = 1.17 [ms], TS_2 = 0.74 [ms], and a value expressing the dependence of 0.98 ms, differs in minimum initial index, indicating the need to locate the service within its limits; in addition, this unit has the required computing resources.

The schematic diagram shown in Figure 9 provides a visualization of the information calculated for all fog equipment units. All micro-sectors can correspond to the required quality of service. An example of such fragmentation, where micro-sectors offer a preset quality of service, is represented by beige ovals. An example of the use of such sectors can be seen in Figure 9, which shows micro-sectors that share a common quality feature. Deviation from this conditional value resulted in only minor changes in the distance to the cluster centroid (less than 5%), which allows the proposed method to be applied to small deviations in cluster types.

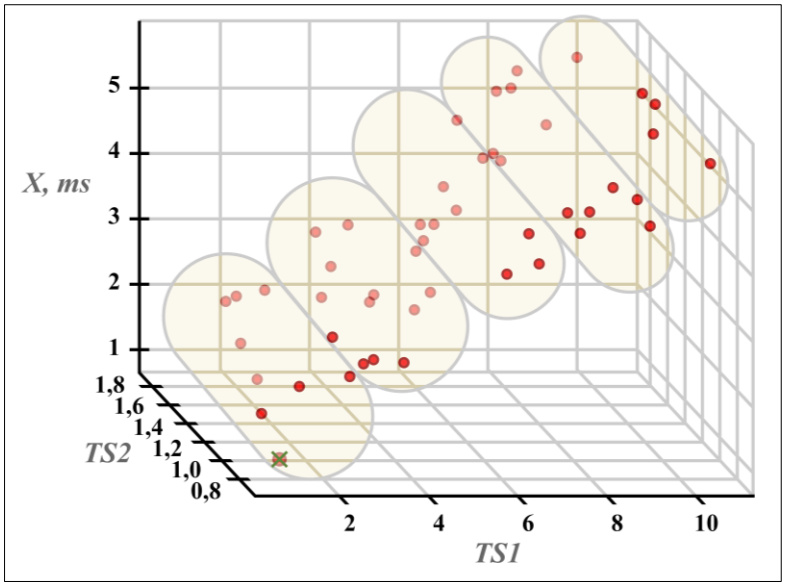


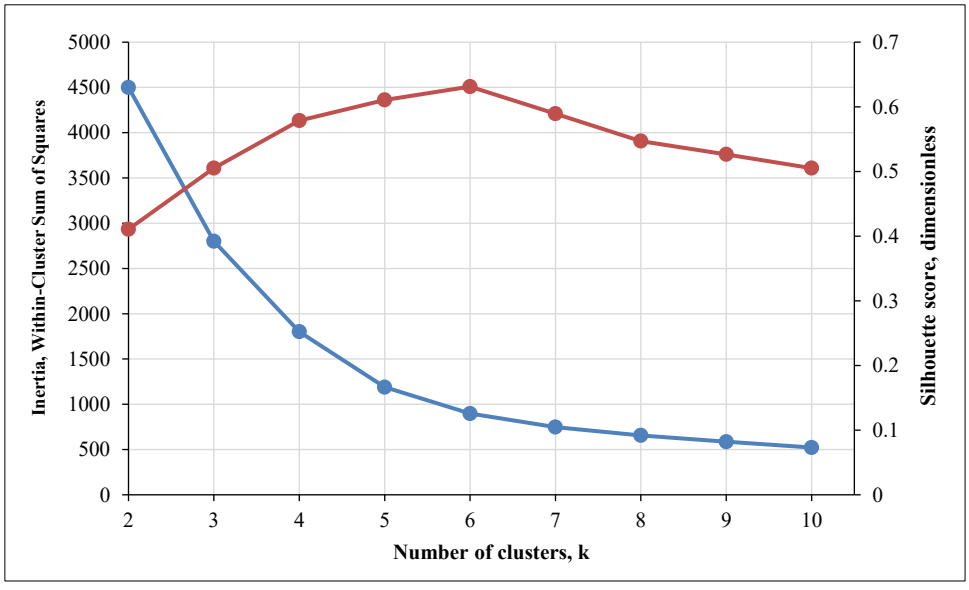
Figure 9. Fragmentation of the sector within which fog computing is performed across service sectors

The aforementioned fragmentation is merely virtual in terms of monitoring and adjusting the proposed concept. It is imperative to organize the operational functioning of the methods responsible for selecting fog equipment in specific fog sectors to ensure the migration of services and maintain the previously specified level of service quality.

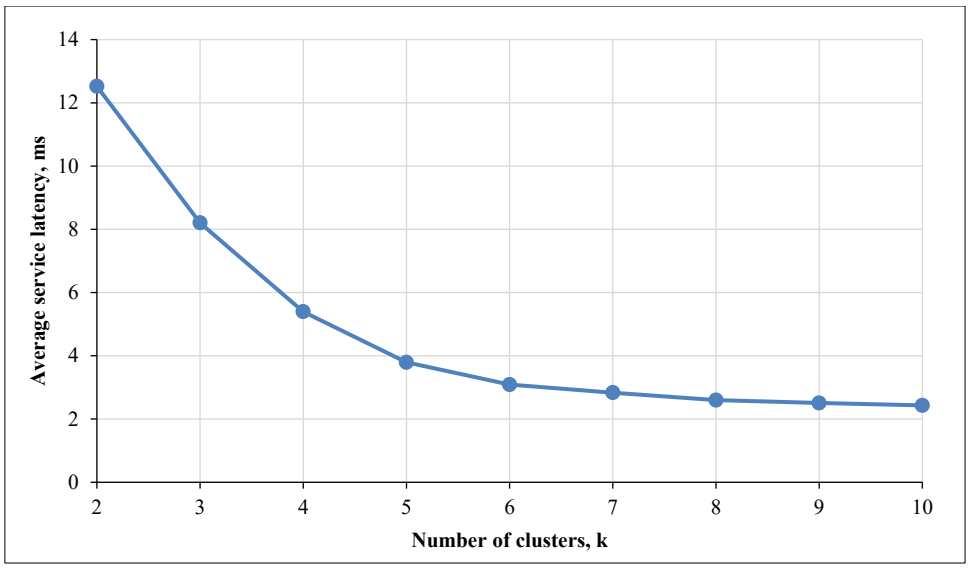
We evaluated DBSCAN as an alternative clustering approach to assess whether non-spherical clusters could improve service placement. However, DBSCAN identified only 3 major clusters in our topology, merging users from different floors (clusters 2 and 5) due to their spatial proximity in the horizontal plane, despite significant vertical separation. This resulted in 18% increase in average TS_1 compared to K-means, as fog nodes were placed at suboptimal heights.

4-2- Clustering Sensitivity Study

To ascertain the optimal number of user clusters, k, for the division of the fog sector, a sensitivity analysis was conducted. The Elbow method and Silhouette Score analysis was utilized (Figure 10a), which demonstrated that k=6 provides the optimal balance between cluster compactness and separation quality. The presence of inertia is evident at k=6, where the additional clusters results in a decline in the return on reducing the dispersion within the cluster. Concurrently, the Silhouette Score attains its maximum at k=6 (0.63), thereby validating the optimal cohesion of the cluster in relation to its neighboring clusters. As demonstrated in Figure 11b, the service delay is shown to be significantly reduced from 12.5 milliseconds to 2.8 milliseconds. However, after this point, the improvement becomes insignificant (<10%). This finding serves to substantiate the observation that the enhancement in performance that was noted in the preliminary experiments is indicative of the efficacy of spatial clustering.



(a)



(b)

Figure 10. Clustering sensitivity analysis: a) Cluster validity analysis, b) Impact of cluster count on latency

To ascertain the optimal number of fog sectors, a sensitivity analysis was conducted using both the Elbow method and Silhouette analysis. It has been demonstrated that increasing k has a beneficial effect on inertia and delay, but this effect is counteracted by a decline in performance at k>6. Furthermore, the Silhouette score attains its maximum at k=6 (0.63), thus indicating the most distinct separation of user groups. The selection of k > 6 results in a marginal enhancement of latency (less than 0.3 milliseconds) accompanied by a substantial escalation in routing intricacy and overhead, which is shown on Figure 10.

4-3- PSO Sensitivity Analysis

The selection of PSO algorithm parameters was based on canonical convergence analysis. The approach employed utilized a decay coefficient in order to circumvent the occurrence of premature convergence. The inertia weight was set to 0.7298. The swarm size was fixed at 30, and the number of iterations was set to 100, as previous research showed that convergence stabilizes within 50–70 iterations.

Figure 11 shows the sensitivity of the proposed method to changes in inertial weight. The canonical value of 0.729 gives the smallest finite delay and stable convergence. Lower values lead to premature convergence at suboptimal local minima, while higher values lead to excessive oscillations and slower convergence.

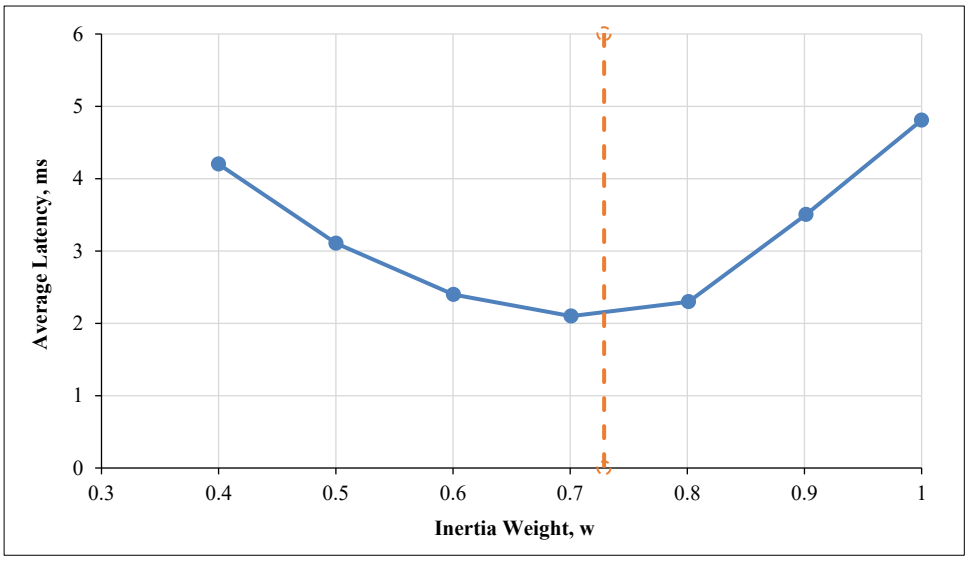


Figure 11. Inertia weight impact on service latency

4-4- Monitoring the Compilation of Forecasts Regarding Controller Load in SDN

In order to validate the selected set of characteristics, a number of RNN configurations are evaluated with different input data. The base model utilized solely traffic volume indicators (ByteCount, PacketCount), whereas the extended variants incorporated CPU and RAM usage, in addition to supplementary metrics, as illustrated in Table 1.

Table 1. Impact of RNN feature sets on load prediction accuracy

RNN feature set	Input features description	RMSE	MAE
A (baseline)	ByteCount, PacketCount	0.00387	0.00291
B	A set with CPU and RAM	0.00345	0.00264
C	B set with aggregated flow-level load	0.00339	0.00258

As demonstrated in Table 1, the incorporation of resource metrics into the fundamental feature set of configuration B leads to a modest enhancement in RMSE and MAE. Further enhancement by means of aggregated flow-level load descriptors in configuration C procures merely marginal supplementary benefits, whilst concomitantly escalating the load on data collection and processing in ultra-low latency environments. This finding validates the hypothesis that a compact feature set based on ByteCount, PacketCount, CPU and RAM provides a trade-off between accuracy and complexity for load prediction.

To calculate the characteristics required for the function describing the studied indicators, a special software package was created. This software package initially generated an information package and then calculated the characteristics of the correlation moment and the coefficients ξX and ρU. The calculation of these coefficients enables the determination of the parameters that affect the processor and RAM load and demonstrates the ex-tent to which they do so.

The theoretical basis for selecting this methodology was the interdependence of measured values with independent ones. Consequently, as demonstrated in Figure 12, it is challenging to analyze the degree of interdependence between the two evaluated characteristics. For this purpose, it was necessary to convert the initial matrix to one of the standardized formats. The z-score standardization method was employed to normalize the data. Subsequently, an altered version of the matrix was employed to construct a scatter plot, thereby illustrating the distribution of weighted scores in the given set.

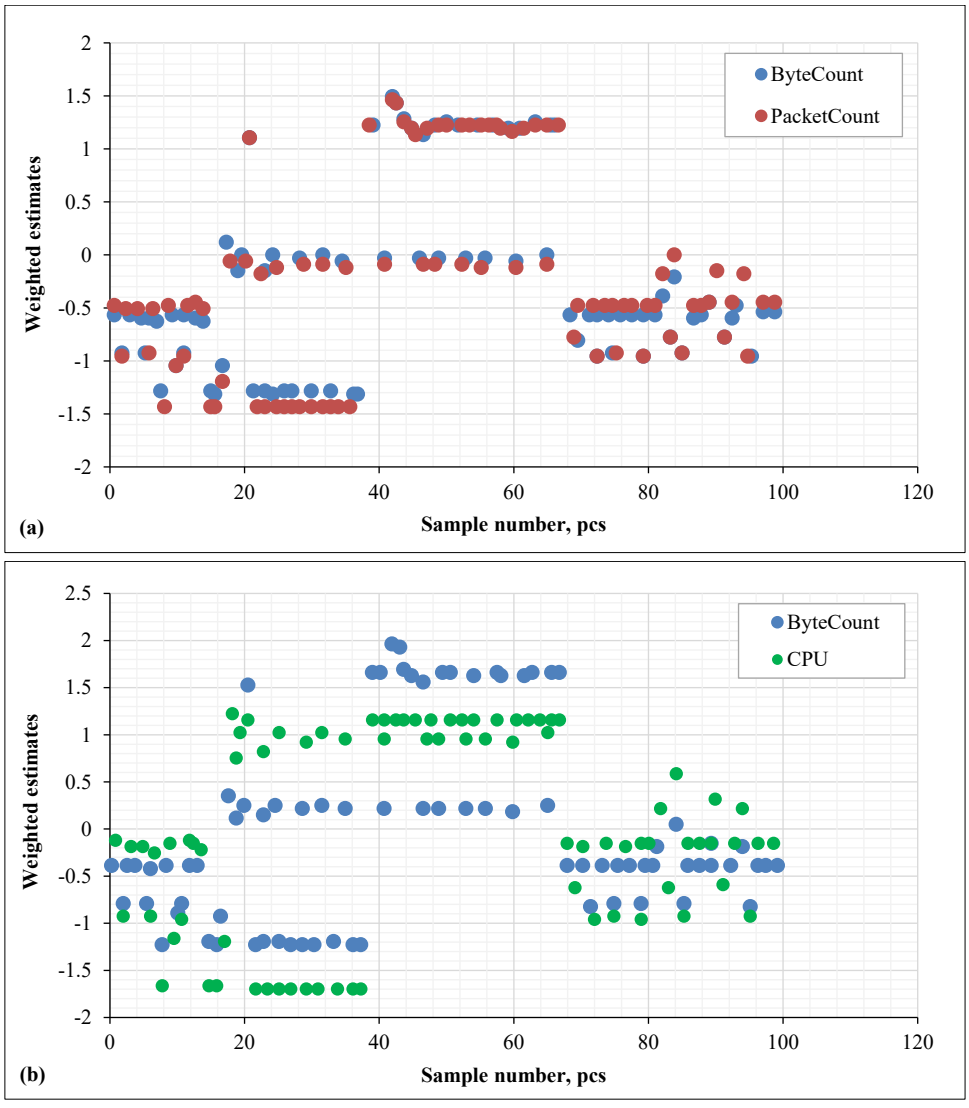


Figure 12. Scatter plot of weighted estimates distribution: a) Relationship between ByteCount and Pack-etCount, b) Relationship between ByteCount and CPU

As illustrated in Figure 12, the comprehensive list encompasses all characteristics indicative of a correlation between CPU load and ByteCount. The total amount of data subjected to testing was 240 time series of network traffic. As demonstrated in Diagram a, there is a clear correlation between two fundamental characteristics that facilitate the analysis of the desired indicators. As demonstrated in Diagram b, the distribution of the weighted estimates of the indicators exhibits distinct variations at this stage. It is imperative to analyze these parameters considering the function that encompasses all potential alterations.

To assess the level of mutual correlation between the indicators under study, it is necessary to calculate the mutual correlation index. Each calculated value of the index must be fully compliant with the stipulated conditions, namely $|\rho jk|>1$ and $|\rho jk|<1$. This observation signifies the existence of a symbiotic relationship between the analyzed indicators. As the mutual connection indicator increases, it becomes possible to predict a change in the current value of the second indicator. To proceed with the calculation of numerous values, it is first necessary to determine the indicator of mutual connection between ByteCount and CPU, corresponding to 0.88. Such values indicate a strong correlation between traffic volumes and processor load. The desired value is instrumental in determining the dependence of the OpenFlow service flow on the load of the SDN controller.

An additional analysis of the correlation between ByteCount-CPU was also performed, as shown in Table 2. It is evident that ByteCount-CPU exhibits a robust correlation ($\rho > 0.79$, $p < 0.001$) across diverse traffic models, network scales, and protocol sets. A slight deterioration in correlation during peak loads is indicative of temporary queuing effects.

Table 2. ByteCount-CPU correlation stability across different scenarios

Scenario	Traffic pattern	Correlation index ρ	p-value
Baseline	UDP	0.88	<0.001
Peak	Bursty HTTP	0.82	
Mixed	TCP + UDP	0.79	
Low load	IoT sensors	0.85	

Consequently, to conclude regarding the functionality or load of the device, it is necessary to form an analytical simulation that summarizes all internal OpenFlow streams by programmatically altering the network configuration. The calculated list of DataSet ML facilitates the construction of a graph representing the spread of the analyzed values (Figure 13).

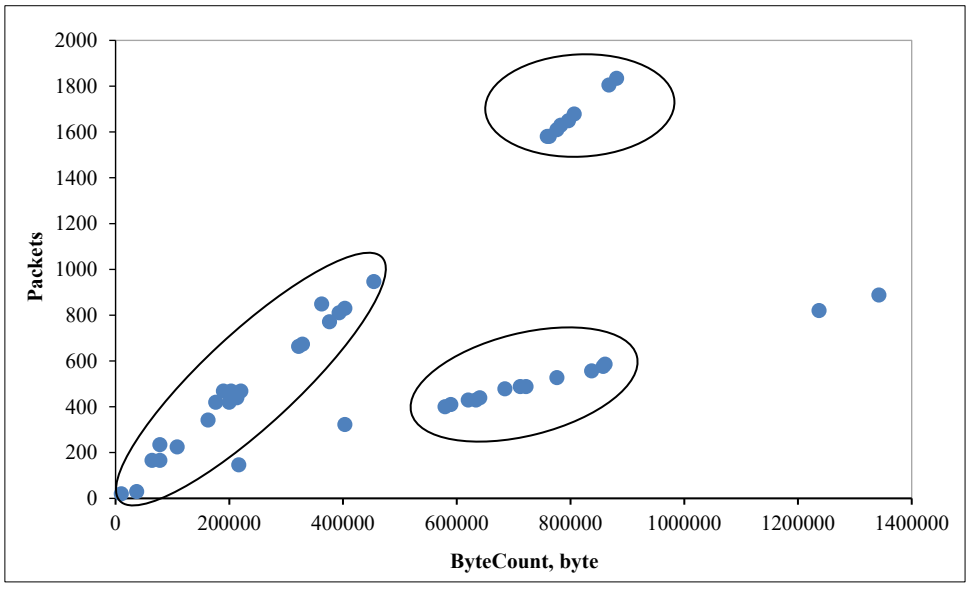


Figure 13. External view of a scatter plot describing the distribution of values that are part of the DataSet ML list

When training the neural network, its performance was analyzed using the MSE indicator shown in formula 11, i.e., the mean square error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (11)$$

where Y_i is the gradient of the detected values of the theoretically-predicted argument, $\hat{Y}_i$ is the gradient of the predicted values, n is a number of examples in the sample and the MSE value is the mean prediction error normalized before the calculation.

Thus, the selected neural network with its architecture and parameters provides the traffic forecast MSE_{Train}=4.54*10^6; MSE_{Test}=1.5*10^6. Figure 14 shows the change in the MSE parameter and indicates the power trend lines based on the training and test datasets.

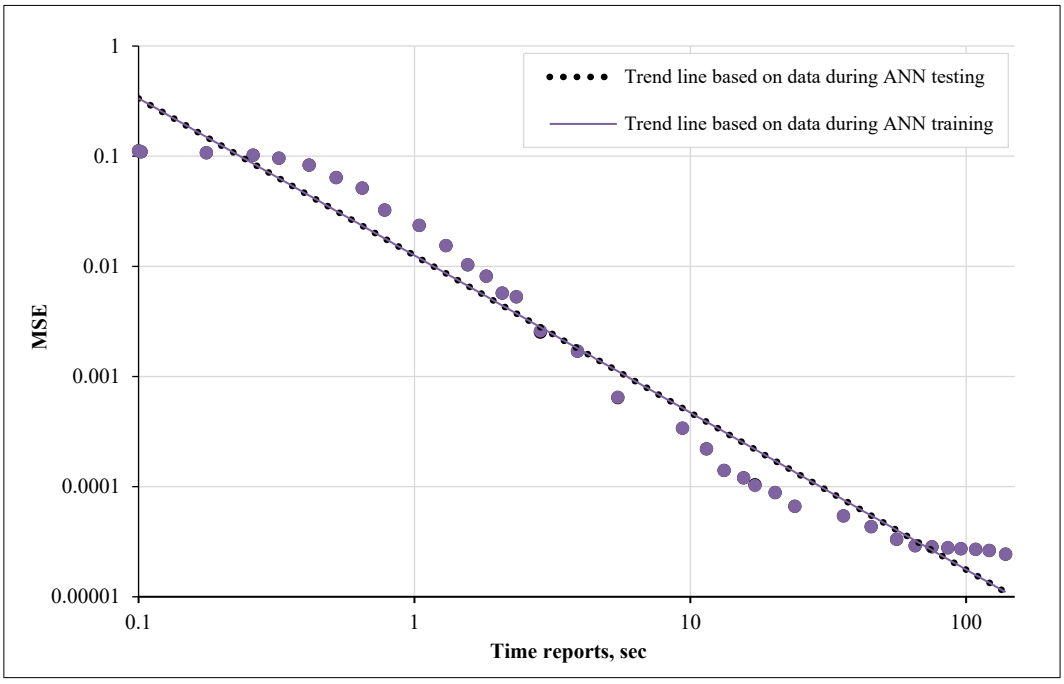


Figure 14. Power estimation of the training and testing samples

5- Discussion

The combination of user clustering, optimized node selection, and load forecasting has been shown to significantly reduce load by adjusting the distribution of services relative to user groups and the predicted server load. This phenomenon is especially evident during periods of high network load because predictive migration facilitates the avoidance of peak load values. The findings indicated a correlation between the total volume of data and the CPU load, suggesting that the data obtained through OpenFlow is adequate for evaluating SDN performance. The proposed four-level recurrent network provides a competitive forecast.

A comparative analysis of the results with those presented in the most relevant con-temporary studies reveals the competitiveness of the proposed method. In analogous studies on analogous tasks, with samples that are approximately similar or slightly smaller in volume, it is possible to find MSE values ranging from 0.000089 to 0.001. The observed variations in the values can be attributed to the varying forecast horizons of the papers and the distinct normalization procedures employed. The findings corroborate the efficacy of RNN in addressing telecommunication challenges, including the dynamic de-centralized analysis of traffic movement patterns and service orchestration integration [31, 32].

The accuracy of SDN controller load prediction using RNN in our study reached 0.00387 RMSE, which outperforms many similar studies in the field of fog computing. A study by Jin & Rezaeipanah [33], using LSTM to predict fog device failures, demonstrated an accuracy of 98.69% on test data with a normalized RMSE of 0.017. This approach focuses only on predicting hardware failures, unlike the proposed method, which focuses on predicting microservice migrations that lead to SLA violations. The Root Mean Square Error (RMSE) achieved for controller load prediction is comparable to that of analogous scientific works, such as the LSTM-based failure prediction model. While the aforementioned work focuses on predicting device failures, the present model aims to make proactive migration decisions to prevent Service Level Agreements (SLAs) from being violated using only ByteCount and PacketCount instead of deep packet inspection.

The integration of K-means methods with PSO optimization demonstrates significant advantages over modern similar methods. For example, Lin et al. [34] proposes using an optimized service placement policy based on a metaheuristic algorithm. However, this approach does not take into account spatial clustering of users, leading to suboptimal resource allocation. The findings indicate that integrating Fog computing with traffic forecasting enables a substantial reduction in load during service. In contrast to analogous studies in the domain of contemporary literature, the proposed assessment aims to migrate microservices based on load forecasting to an SDN controller.

The forecasting model presented here demonstrates that the parameters of the number of clusters, the PSO algorithm settings, and the length of the neural network input have a significant impact on the final quality parameters. The model uses fixed weights to ensure computational simplicity and adherence to SLA parameters, but the integration of adaptive

weight values could theoretically improve the performance of the proposed algorithm, as modern research shows that dynamically adjusting the weights of fitness functions in heterogeneous Fog computing in the context of ultra-low latency can improve the convergence of the considered timing characteristics [35]. The optimal number of clusters is crucial for accurately identifying segments and distributing users across them efficiently. The PSO parameters determine the convergence of nodes, affecting accuracy and time. Note that the length of the RNN input is of paramount importance, as it facilitates precise prediction of load dynamics. As a direction for future research, it is worth testing the proposed methodology on a different data set, namely traffic patterns, changing protocol combinations and data volumes, which will determine how this methodology will perform in different contexts, each of which will be useful for a specific area of application.

6- Conclusions

A novel solution to the prevailing issues in the communication of distributed computing and the support of associated services by fifth-generation communication networks is hereby proposed. The concept under consideration focuses on the specifics of microservice communication, which is designed to minimize the distance between the client and the service. The methodology for dynamic redistribution of fog equipment combines user clustering and load forecasting using recurrent neural networks and node selection optimization based on PSO. A substantial decrease in request processing time, reaching up to 69% reduction, has been observed, exhibiting high dynamics.

To confirm the assumption regarding the forecasting of computing resource work-load, a verification methodology based on multifactorial assessment of mutual relation-ships is proposed. The findings of the calculation of the requisite indicators demonstrate that internal communication activity exhibits a non-linear relationship with SDN load. To forecast the load on a device that programmatically controls the network configuration, it is proposed to use a recurrent neural network with a structure consisting of four nested levels.

The developed hybrid algorithmic process is distinguished by its decomposed objective function, which optimizes the position of objects and the computational process. The proposed formalization of adaptive triggers constitutes an additional method of adaptive control applicable to various tasks in distribution systems. The proposed model demonstrates how deep learning and heuristic optimization methods can be combined to solve the complex problem of migration in distributed networks with high event rates and variable topology. The proposed approach can be utilized by communication service providers and MEC system developers in the development of effective load management mechanisms, thereby ensuring the maintenance of Service Level Agreements (SLAs) in conditions characterized by dense mobility and rapid traffic growth.

The proposed methodology is subject to certain limitations. The experimental evaluation was conducted on simulated data with a fixed fog sector topology and a limited number of simultaneously migrating microservices. The paper is also deficient in its failure to provide an analysis of the impact of individual fog node failures. Furthermore, there has been no implementation of mechanisms for automatic recovery of such nodes. Future research could expand the topology to a much larger number of nodes and replace the fixed weight values with an adaptive objective function. It would also be worthwhile to extend the presented mode to more complex scenarios with changeable topology, integrating new parameters, and fault tolerance.

7- Declarations

7-1- Author Contributions

Conceptualization, V.Zh.K. and E.Yu.L.; methodology, M.M. and N.Z.I.; software, I.A.; validation, I.A.; formal analysis, I.A. and M.M.; investigation, M.M.; resources, I.A. and M.M.; data curation, M.M.; writing—original draft preparation, M.M.; writing—review and editing, I.A. and M.M.; visualization, N.Z.I.; supervision, E.Yu.L.; project administration, E.Yu.L.; funding acquisition, V.Zh.K. All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7-3- Funding

This work was supported by a grant, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4G0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated June 20, 2025 No. 139-15-2025-011.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

[1] Chowdhury, M. Z., Shahjalal, M., Ahmed, S., & Jang, Y. M. (2020). 6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions. IEEE Open Journal of the Communications Society, 1, 957–975. doi:10.1109/OJCOMS.2020.3010270.

[2] Ray, K., Banerjee, A., & Narendra, N. C. (2020). Proactive Microservice Placement and Migration for Mobile Edge Computing. Proceedings - 2020 IEEE/ACM Symposium on Edge Computing, SEC 2020, 28–41. doi:10.1109/SEC50012.2020.00010.

[3] Parvez, I., Rahmati, A., Guvenc, I., Sarwat, A. I., & Dai, H. (2018). A survey on low latency towards 5G: RAN, core network and caching solutions. IEEE Communications Surveys and Tutorials, 20(4), 3098–3130. doi:10.1109/COMST.2018.2841349.

[4] Srisamarn, U., Pradittasnee, L., & Kitsuwan, N. (2021). Resolving Load Imbalance State for SDN by Minimizing Maximum Load of Controllers. Journal of Network and Systems Management, 29(4), 46. doi:10.1007/s10922-021-09612-w.

[5] Zhang, C., Patras, P., & Haddadi, H. (2019). Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys and Tutorials, 21(3), 2224–2287. doi:10.1109/COMST.2019.2904897.

[6] Zhu, J., Chen, H., & Wang, H. (2025). SDT-MCS: Topology-Aware Microservice Orchestration with Adaptive Learning in Cloud-Edge Environments. Concurrency and Computation: Practice and Experience, 37(18–20), 70176. doi:10.1002/cpe.70176.

[7] Rejiba, Z., Masip-Bruin, X., & Marín-Tordera, E. (2020). A survey on mobility-induced service migration in the fog, edge, and related computing paradigms. ACM Computing Surveys, 52(5), 1–33. doi:10.1145/3326540.

[8] Bellavista, P., Corradi, A., Foschini, L., & Scotece, D. (2019). Differentiated service/data migration for edge services leveraging container characteristics. IEEE Access, 7, 139746–139758. doi:10.1109/ACCESS.2019.2943848.

[9] Mahmoud, M., Ashraf Ateya, A., Muthanna, A., Zaghloul, A., Kirichek, R., & Koucheryavy, A. (2021). Distributed Edge Computing to Assist LPWAN: Fog-MEC Model. ACM International Conference Proceeding Series, 587–594. doi:10.1145/3508072.3508192.

[10] Toumi, N., Bagaa, M., & Ksentini, A. (2023). Machine Learning for Service Migration: A Survey. IEEE Communications Surveys and Tutorials, 25(3), 1991–2020. doi:10.1109/COMST.2023.3273121.

[11] Pallewatta, S., Kostakos, V., & Buyya, R. (2023). Placement of Microservices-based IoT Applications in Fog Computing: A Taxonomy and Future Directions. ACM Computing Surveys, 55(14 S), 1–43. doi:10.1145/3592598.

[12] Taleb, I., Guillaume, J. L., & Duthil, B. (2025). A Survey on Services Placement Algorithms in Integrated Cloud-Fog / Edge Computing. ACM Computing Surveys, 57(11), 1–36. doi:10.1145/3729214.

[13] Jasim, M., & Siasi, N. (2024). Local Load Migration in High-Capacity Fog Computing. ACM Transactions on Internet Technology, 24(4), 1–31. doi:10.1145/3690386.

[14] Barbarulo, F., Puliafito, C., Virdis, A., & Mingozzi, E. (2022). Extending ETSI MEC Towards Stateful Application Relocation Based on Container Migration. Proceedings - 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2022, 367–376. doi:10.1109/WoWMoM54355.2022.00035.

[15] Escolar, A. M., Alcaraz-Calero, J. M., Salva-Garcia, P., Bernabe, J. B., & Wang, Q. (2021). Adaptive Network Slicing in Multi-Tenant 5G IoT Networks. IEEE Access, 9, 14048–14069. doi:10.1109/ACCESS.2021.3051940.

[16] Farooq, M. S., Riaz, S., & Alvi, A. (2023). Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review. Electronics (Switzerland), 12(14), 3077. doi:10.3390/electronics12143077.

[17] Kerimkhulle, S., Dildebayeva, Z., Tokhmetov, A., Amirova, A., Tussupov, J., Makhazhanova, U., Adalbek, A., Taberkhan, R., Zakirova, A., & Salykbayeva, A. (2023). Fuzzy Logic and Its Application in the Assessment of Information Security Risk of Industrial Internet of Things. Symmetry, 15(10). doi:10.3390/sym15101958.

[18] Troia, S., Martinez, D. E., Martin, I., Zorello, L. M. M., Maier, G., Hernandez, J. A., Gonzalez De Dios, O., Garrich, M., Romero-Gazquez, J. L., Moreno-Muro, F. J., Marino, P. P., & Casellas, R. (2019). Machine Learning-assisted Planning and Provisioning for SDN/NFV-enabled Metropolitan Networks. 2019 European Conference on Networks and Communications, EuCNC 2019, 438–442. doi:10.1109/EuCNC.2019.8801956.

[19] Le, L. V., Sinh, D., Lin, B. S. P., & Tung, L. P. (2018). Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management. 2018 4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018, 207–211. doi:10.1109/NETSOFT.2018.8460129.

[20] Jiang, W., Han, H., He, M., & Gu, W. (2024). ML-based pre-deployment SDN performance prediction with neural network boosting regression. Expert Systems with Applications, 241, 122774. doi:10.1016/j.eswa.2023.122774.

[21] Aouedi, O., Le, V. A., Piamrat, K., & Yusheng, J. I. (2025). Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions. ACM Computing Surveys, 57(6), 1–37. doi:10.1145/3703447.

[22] Ogundoyin, S. O., & Kamil, I. A. (2023). Optimal fog node selection based on hybrid particle swarm optimization and firefly algorithm in dynamic fog computing services. Engineering Applications of Artificial Intelligence, 121, 105998. doi:10.1016/j.engappai.2023.105998.

[23] Wu, Z., Fan, X., Bian, G., Liu, Y., Zhang, X., & Chen, Y. Q. (2025). Short-term wind power forecast with turning weather based on DBSCAN-RFE-LightGBM. Renewable Energy, 251, 123217. doi:10.1016/j.renene.2025.123217.

[24] Sousa, M., Vieira, P., Queluz, M. P., & Rodrigues, A. (2025). Performance Analysis of Mobile Wireless Networks Through Crowdsourcing Data Clustering. 2025 IEEE International Mediterranean Conference on Communications and Networking, MeditCom 2025, 1–6. doi:10.1109/MeditCom64437.2025.11104421.

[25] Aleisa, M. A. (2025). Traffic classification in SDN-based IoT network using two-level fused network with self-adaptive manta ray foraging. Scientific Reports, 15(1). doi:10.1038/s41598-024-84775-5.

[26] v, J., & Akilandeswari, J. (2025). AI-Driven Load Balancing for Scalable Software Defined Network (SDN) Multi-Controller Architectures. International Journal of Computer Networks and Applications, 12(6), 862–878. doi:10.22247/ijcna/2025/51.

[27] Jiang, L. (2025). A Network Anomaly Traffic Detection Method Based on CNN-LSTM. Security and Privacy, 8(3), 70033. doi:10.1002/spy2.70033.

[28] Gill, S. S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., Golec, M., Stankovski, V., Wu, H., Abraham, A., Singh, M., Mehta, H., Ghosh, S. K., Baker, T., Parlikad, A. K., Lutfiyya, H., Kanhere, S. S., Sakellariou, R., Dustdar, S., … Uhlig, S. (2022). AI for next generation computing: Emerging trends and future directions. Internet of Things (Netherlands), 19, 100514. doi:10.1016/j.iot.2022.100514.

[29] Vaño, R., Lacalle, I., Sowiński, P., S-Julián, R., & Palau, C. E. (2023). Cloud-Native Workload Orchestration at the Edge: A Deployment Review and Future Directions. Sensors, 23(4), 2215. doi:10.3390/s23042215.

[30] Yang, Y., Geng, S., Zhang, B., Zhang, J., Wang, Z., Zhang, Y., & Doermann, D. (2023). Long term 5G network traffic forecasting via modeling non-stationarity with deep learning. Communications Engineering, 2(1), 33. doi:10.1038/s44172-023-00081-4.

[31] Kablaoui, R., Ahmad, I., Abed, S., & Awad, M. (2024). Network traffic prediction by learning time series as images. Engineering Science and Technology, an International Journal, 55, 101754. doi:10.1016/j.jestch.2024.101754.

[32] Dodan, M. E., Vien, Q. T., & Nguyen, T. T. (2022). Internet Traffic Prediction Using Recurrent Neural Networks. EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 9(4), 1. doi:10.4108/eetinis.v9i4.1415.

[33] Jin, W., & Rezaeipanah, A. (2025). Dynamic task allocation in fog computing using enhanced fuzzy logic approaches. Scientific Reports, 15(1), 18513. doi:10.1038/s41598-025-03621-4.

[34] Lin, Y., Shi, Y., & Mohammadnezhad, N. (2024). Optimized dynamic service placement for enhanced scheduling in fog-edge computing environments. Sustainable Computing: Informatics and Systems, 44, 101037. doi:10.1016/j.suscom.2024.101037.

[35] Saad, M., Enam, R. N., & Qureshi, R. (2024). Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application. Frontiers in Big Data, 7. doi:10.3389/fdata.2024.1358486.