Mathematical and knowledge-oriented modeling of efficient and balanced operation of a university computer network

Andriy Melnyk[1,5,†], Serhiy Voznyak[1,*,†], Svitlana Sachenko[1,†], Volodymyr Naumchuk[2,†] and Oleh Korkushko[4,†]

[1] *West Ukrainian National University, Lvivska Street 11, 46001 Ternopil, Ukraine*

[2] *Ternopil Volodymyr Hnatiuk National Pedagogical University, Maxyma Kryvonosa Street 2, 46027 Ternopil, Ukraine*

[3] *Catholic University in Ruzomberok Ruzomberok, Hrabovská cesta 1A, 034 01 Ruzomberok, Slovakia*

[4] *Educational and rehabilitation institute of higher education «Kamianets-Podilskyi State Institute», Godovantsya Street 13, 32300 Kamianets-Podilskyi, Ukraine*

[5] *Department of Clinical Engineering, Academy of Silesia, 40-555 Katowice, Poland*

Abstract

This study presents a mathematical and knowledge-oriented framework for analyzing and optimizing the efficient and balanced operation of a university computer network. The growing complexity of higher education ICT infrastructures, the expansion of digital services, and the intensification of user activity require advanced modeling techniques capable of supporting data-driven management decisions. The proposed approach integrates mathematical modeling, performance analysis, and knowledge-based methods to evaluate structural balance, operational efficiency, and resource utilization within institutional networks. A key contribution of this work is the development of a dynamic load-distribution model for terminal cluster centers, which are responsible for processing high-intensity user requests in academic environments. The model incorporates temporal and structural characteristics of network traffic, adaptive balancing strategies, and knowledge-driven rules for predicting load fluctuations across distributed terminal clusters. This enables the system to reallocate computational resources in real time, prevent overload states, and maintain stable quality-of-service indicators under varying workloads. The results demonstrate that combining mathematical modeling with knowledge-oriented decision mechanisms significantly enhances network efficiency, reduces response delays, and ensures balanced utilization of computational and communication resources. The proposed framework can serve as a basis for designing intelligent management systems for university ICT infrastructures and contributes to the development of advanced methods for performance optimization in educational networks.

Keywords

Mathematical modeling, knowledge-oriented approach, university computer networks, network efficiency, load balancing, terminal clusters, resource optimization, knowledge-based systems, performance analysis.

1. Introduction

Modern corporate networks of higher education institutions are characterized by a high degree of distribution, intensive traffic, and extensive use of terminal servers for providing access to information resources, virtual laboratories, and educational services. Under such conditions, ensuring reliable user identification and authentication, as well as maintaining uninterrupted operation of network communication channels, becomes critically important. Most university infrastructures rely on Kerberos-based technologies, which establish distributed authentication systems and require consistent interaction among terminal servers [1–3].

AIT&AIS'2025: International Scientific Workshop on Applied Information Technologies and Artificial Intelligence Systems, December 18–19 2025, Chernivtsi, Ukraine

* Corresponding author.

† These authors contributed equally.

ame@wunu.edu.ua (A. Melnyk); sv@wunu.edu.ua (S. Voznyak); v_i_n@tnpu.edu.ua (V. Naumchuk); s.sachenko@wunu.edu.ua (S. Sachenko); oleg-ua82@ukr.net (O. Korkushko)

0000-0001-7799-9877 (A. Melnyk); 0000-0000-0000-0000 (S. Voznyak); 0000-0002-2919-0720 (V. Naumchuk); 0000-0001-8225-1820 (S. Sachenko); 0000-0001-6577-8647 (O. Korkushko)

Despite the widespread adoption of terminal networks, several essential issues remain unresolved. In particular, the optimal load distribution among terminal servers and the assurance of communication channel survivability under failures, peak loads, or uneven resource utilization constitute some of the most urgent challenges in the network infrastructures of modern universities. Insufficient fault tolerance and ineffective load balancing may lead to prolonged authentication delays, reduced service availability, and impaired performance of educational and research platforms [4, 5].

Therefore, the development of models, methods, and algorithms aimed at enhancing communication channel survivability, optimizing the exchange of authentication information, and enabling dynamic load distribution among terminal servers represents a relevant scientific task. Addressing these issues will contribute to improving the resilience, scalability, and efficiency of corporate networks in higher education institutions, in line with contemporary trends in the development of secure and highly available information and communication systems [6, 7].

2. Task Statement

Corporate computer networks in higher education institutions extensively rely on terminal servers and Kerberos-based infrastructure to provide user authentication and access to educational and scientific services. In such systems, user groups are attached to specific terminal servers, which interact with one another on a peer-to-peer basis and form a distributed authentication subsystem. The reliability and efficiency of this subsystem depend on its ability to maintain minimal authentication service time and preserve service availability even in the presence of server failures [8, 9].

In real operational environments, terminal servers may fail, become overloaded, or operate in degraded mode. Under such conditions, users of failed servers must be promptly reassigned to other functional servers, taking into account available resources, load levels, access policies, network topology, switching costs, and performance constraints. At the mathematical level, this problem is formulated as the redistribution of user groups among terminal servers while minimizing the average service time and satisfying constraints related to flow intensities, memory resources, security policies, and communication bandwidth [10].

However, exhaustive enumeration of all possible redistribution variants is computationally infeasible: the number of alternatives grows exponentially with the number of servers. Therefore, solving this problem requires specialized methods for reducing the search space and developing efficient optimization algorithms. Traditional mathematical models do not incorporate logical, policy-based, and semantic dependencies among network components, which limits the practical relevance of their outputs [11].

At the same time, existing terminal network management systems do not employ semantic technologies to represent knowledge about the infrastructure and thus cannot produce context-aware redistribution decisions. Consequently, the research problem is to develop an integrated knowledge-oriented model for load redistribution in terminal cluster centers, which combines: a formal stochastic model of the terminal network; an ontology that captures structural, operational, and policy constraints, an optimization algorithm that accounts for both numerical parameters and semantic dependencies, semantic filtering methods based on SPARQL queries, KnowledgeRule specifications, and logical reasoning.

The objective of the study is to determine an optimal plan for reassigning users from failed servers to operational servers while minimizing the average service time and switching cost under technical and semantic constraints.

3. Modeling the Dynamic Load Distribution of Terminal Cluster Centers

One of the key challenges in organizing communication channels among terminal servers in corporate university networks is ensuring reliable identification and authentication of a closed group of authorized users, (Fig. 1). In modern infrastructures, user identification and authentication during the establishment of network communication channels are predominantly implemented through Kerberos-based mechanisms built on the client–server paradigm [12–14]. According to this approach, terminal networks are partitioned into Kerberos realms, each containing a dedicated authentication server responsible for granting authorized users controlled access to approved information resources.

Terminal servers interact with each other through pairwise communication channels, forming a distributed authentication system. These servers share a common secret key and exchange authentication information required for validating user identities within the system. Each server maintains a localized database that stores credentials and authorization attributes of legitimate users.

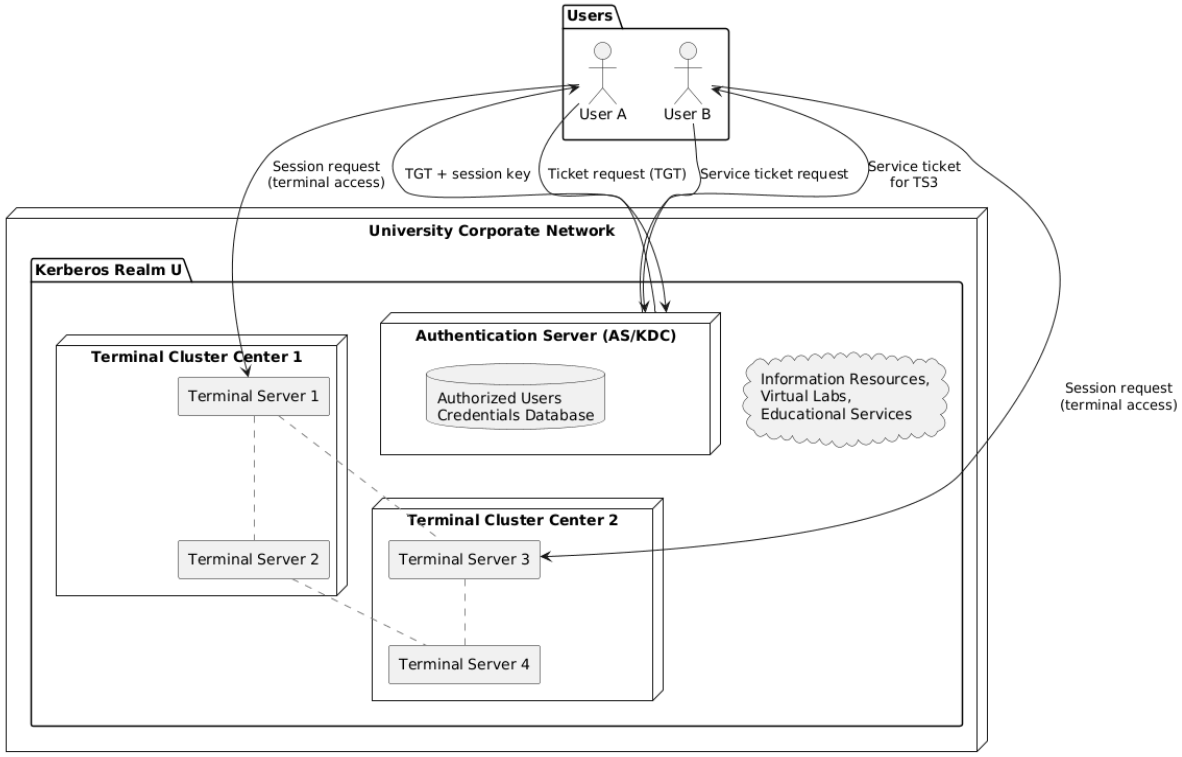

Figure 1: University corporate network with terminal cluster centers and Kerberos authentication.

It should be emphasized that, despite the increasing interest in terminal-oriented network technologies, numerous issues related to their practical organization, scalability, and reliability remain insufficiently addressed [15]. One of the most significant challenges in this domain is the development of efficient methods and algorithms aimed at improving the survivability of communication channels. Such mechanisms must guarantee that authorized users retain timely access to requested resources even under various failure scenarios, including partial server outage, link degradation, or unexpected load surges [16, 17].

To address these challenges, the modeling of dynamic load distribution among terminal cluster centers becomes a critical research direction. At the initial stage of this modeling process, the structure of the computer network is formally represented using an open stochastic network, which enables the analytical description of probabilistic interactions, load fluctuations, and state transitions occurring in distributed authentication environments.

Let us assume that the set of terminal servers created within the computer network is denoted by $S = \{S_1, S_2, ..., S_n\}$ and that to each server Si a certain group of authorized users from the set U is assigned. In other words, for each server Si there exists a subset $U_i \subset U$ such that the family $\{U_i\}_{i=1}^{n}$ forms a partition of U, i.e.

$$U_i \cap U_j = \emptyset, \quad i \neq j. \tag{1}$$

Each subset U_i consists of individual users u_{ki}. Thus, we can write

$$U = \cup_{i=1}^{n} U_i, \quad [U_i] = n_i, \quad \sum_{i=1}^{n} n_i = m, \quad m > n, \tag{2}$$

where m is the total number of authorized users and n is the number of terminal servers.

Naturally, all servers from the set S perform identical authentication functions, since they implement a unified security policy of the terminal network. As a performance criterion for the operation of the communication channels during authentication, we consider T_0, the average service time of user authentication requests under the condition that all servers from S are fully operational.

Assume further that the terminal network is represented as an exponential open stochastic network composed of a finite number of single-channel queuing systems. These systems form service nodes characterized by a constant arrival intensity λ_0, which does not depend on the network state, at the output of the request source S_0.

Let the intensity λ_0, be known and considered as a parameter of the network. Requests from the source S_0 enter the network with a constant probability p_{0i} of being routed to the queuing system (*QS*) S_i. Requests served by *QS* S_i are then forwarded with a constant probability P_{ij} to *QS* S_i, $j = 1, ..., n$ or leave the network (for $j = 0$), i.e., are returned to the request source. Obviously, the following normalization condition must hold: the sum of the routing probabilities from node S_i over all possible destinations $j = 0, 1, ..., n$ is equal to one.

We now consider the transformation of the input request flow with intensity λ_0 into the input flows of the constituent QSs of the network in the steady-state regime. Let α_i denote the transmission (transformation) coefficient of the input request flow to the input of *QS* S_i, quantitatively equal to the average number of occurrences of an arbitrary request from the network input flow within the input flow of *QS* S_i. Then the intensity of the input flow to *QS* S_i can be expressed in terms of λ_0 as

$$\lambda_i = \alpha_i i \lambda_0. \tag{3}$$

On the other hand, by definition, the fraction of clients from the subset U_i in the total intensity λ_0 can be expressed through the individual request intensities l_{kii} of user u_{ki} directed to the server – *QS* S_i. Extending this relation to all subsets U_i, $i = 1$, we obtain a set of relations connecting the global input intensity λ_0 with the input intensities λ_i of all QSs of the network.

Since a lossless network is considered, the output intensities of the flows from QSs S_i, $i = 1, ..., n$ coincide with the intensities of their input flows. The input intensity of the flow to QSs S_j, $j = 1, ..., n$, is equal to the sum of the flow fraction arriving directly from the request source and the fractions of flows routed from other QSs of the network according to the corresponding routing probabilities.

Taking into account the above relations and the equality $\lambda_i = \alpha_i \lambda_0$, we transform the corresponding balance equation into the following system of linear non-homogeneous algebraic equations with respect to the transmission coefficients α_i, $i = 1, ..., n$, which has a unique solution.

$$\alpha_i = \frac{\sum_{kj=1}^{n} \lambda_{kii}}{\sum_{j=1}^{n} \sum_{kj=1}^{ni} \lambda_{kii}} + \sum_{i=1}^{n} \alpha_i \frac{\lambda_{ii}}{\lambda_i}, \quad j = \overline{1, n}. \tag{4}$$

From this solution, we can determine the average service time $T0$ of client requests:

$$T_0 = \sum_{i=1}^{n} \alpha_i t_i, \tag{5}$$

where $t_i(1 / (\mu_i - \lambda_i))$ is the average service time of requests in $QS\ S_i$. According to the problem statement, the service rates are identical, $\mu_i = \mu$ for all Qss.

We now proceed to formulate the optimization model, which constitutes the basis of the method for improving the fault tolerance of communication channels during the authentication of authorized users.

Let the states of all servers from the set S be defined by the state vector

$$x(k) = \langle x_1(k), \ldots, x_i(k), \ldots, x_n n(k) \rangle, \tag{6}$$

where $x_i(k) = 0$ if server S_i is operational and $x_i(k) = 1$ otherwise. It is known that the total number of such state vectors is 2^n. Among these vectors, we are not interested in the state $\langle 0, 0, \ldots, 0 \rangle$, when all servers are operational, nor in the state $\langle 1, 1, \ldots, 1 \rangle$, when all servers have failed. In other words, we consider only the non-trivial states $x(k)$, $k = 1, \ldots, N$, where $N = 2^n - 2$.

The essence of the problem is as follows: the security administrator of the communication channels, in the presence of failures of some servers corresponding to a state $x(k)$, redistributes their users among the operational servers, subject to certain constraints. For example, since these servers may be geographically distributed, additional costs arise when redirecting users between them.

Assume that the cost required to switch user uki from failed server Si to an operational server S_j in state $x(k)$ is denoted by $C_{ij}^{(k)}$ $i, j = 1, \ldots, ni$. Furthermore, let the memory capacity of the network hardware on which the servers S_i are implemented be given by V_i^{max}, $i = 1, \ldots, n$. The actual memory volume required to host server S_j is denoted by V_j $j = 1, \ldots, n$.

To describe the redistribution of users of failed servers S_i under a given state vector $x(k)$ among the functioning servers S_j, we introduce the pseudo-Boolean variable $x_{ij}(k)$. Here, $x_{ij}(k) = 1$ if the users from the set U_i of the failed server S_i are switched to server S_j for authentication in state $x(k)$, and $x_{ij}(k) = 0$ otherwise. Note that, for each state $x(k)$, all users from U_i are connected to exactly one functioning server.

Based on the above considerations, as well as formulas (2) and (4), we obtain the following optimization model:

$$T_k = \sum_{j=1}^{n} \alpha_i^{(k) t_j(k)} \rightarrow min\, k = \overline{1, N}, \tag{7}$$

where, by applying the expressions for α_i and t_i for the states $x(k)$, we obtain that:

$$\alpha_i(k) = \left[\frac{\sum\limits_{k_j=1}^{ni} \lambda_{kji} + \sum\limits_{i=1}^{n} \sum\limits_{k_i=1}^{ni} \lambda_{ki} X_i^{(k)} X_j^{(k)}}{\sum\limits_{j=1}^{n} \sum\limits_{kj=1}^{nj} \lambda k_j + \sum\limits_{i=1}^{n} \alpha_i^{(k)} \frac{\lambda_{ij}}{\lambda_i} \left(1 - x_i^{(k)} + x_j^{(k)}\right)} \right] \left(1 - x_i^{(k)}\right), \tag{8}$$

$$j = \overline{1, n}, \quad k = \overline{1, N}, \tag{9}$$

with the constraints:

$$t_j^{(k)} = \frac{\left(1 - x_i^{(k)}\right)}{\mu - \left\{ \sum\limits_{kj=1}^{ni} \lambda k_j + \sum\limits_{i=1}^{n} \left[\sum\limits_{k_i=1}^{ni} \lambda k_i x_{ij}^{(k)} + \lambda_{ij} \left(1 - x_j^{(k)}\right) \right] \right\}} \tag{10}$$

$$I = \overline{1, n}, \quad j = \overline{1, n}, \quad k = \overline{1, N}, \tag{11}$$

$$\sum_{i=1}^{n} x_{ij}^{(k)} x_i^{(k)} = n - 1, \quad j = \overline{1, n}, \quad k = \overline{1, N},$$

$$\sum_{i=1}^{n} x_{ij}^{(k)} x_i^{(k)} = 1, \quad i=\overline{1,n}, \quad k=\overline{1,N},$$

$$\sum_{i=1}^{n} \sum_{k=1}^{n\text{i}} \sum_{j=1}^{n} c\, x_{ij}^{(k)} x_i^{(k)} \le C \quad k=\overline{1,N}, \tag{12}$$

$$\sum_{i=1}^{n} V_i x_{ij}^{(k)} x_i^{(k)} \le V_j^{max} - V_j, \quad j=\overline{1,n}, \quad k=\overline{1,N},$$

$$min\left[\mu/\alpha_j^{(k)}\right] > \sum_{i=1}^{n} \sum_{k_i=1}^{n\text{i}} \lambda_{ki}, \quad j=\overline{1,n}, \quad k=\overline{1,N}.$$

Constraint (12) implies that, when redistributing the users of failed servers S_i among the functioning servers S_j under the state vector $x(k)$, the total switching cost must not exceed the predefined threshold C. Inequality (12) imposes an upper bound on the input flow intensity λ_0 under the condition that a steady-state regime exists in the exponential open stochastic network.

As follows from formulas (8)–(12), the algorithm for solving the optimization problem belongs to the class of discrete programming problems with pseudo-Boolean variables. Before developing a practical algorithm suitable for real-world implementation in the design and operation of terminal networks, we first evaluate the computational complexity associated with this model under full enumeration of all possible variants.

It is known that m faulty terminal servers can be selected from n servers in C_n^m different ways. In this case, n - m servers remain operational. According to model conditions (8)–(12), the users of each failed server must be reassigned to one of the (n - m) operational terminal servers. Clearly, for a given number m of failed servers, the total number of possible variants $u(n - m)$ of redistributing them among the (n - m) functioning servers is equal to

$$\Theta(n,m) = C_n^m (n-m)\cdot(n-m)\cdots(n-m) = C_n^m (n-m)^m. \tag{13}$$

Extending this formula to all values of mmm, where $1 \le m \le n$ - 1, we obtain the computational complexity $u(n)$ of the model (9)–(12):

$$u(n) = \sum_{m=1}^{n-1} C_n^m (n-m)^m. \tag{14}$$

It is evident that solving this problem by complete enumeration is practically infeasible [17]. Therefore, when solving such problems, one must aim at an efficient partial enumeration of a comparatively small subset of feasible variants while implicitly pruning the remaining ones.

This objective is addressed by the algorithm corresponding to model (9)–(12), which is based on the branch-and-bound method and takes into account the specific structure of the problem under consideration.

Let us introduce the following notation:

$$I_k = \left\{ i | x_i^{(k)} = 1 \right\} i J_k = \left\{ i | x_i^{(k)} \right\} = 0, \tag{15}$$

where N = {1, 2, ..., n}. Evidently, $I_k = J_k = N/I_k$ that is, the faulty servers form the set I_k, and the functioning servers form the complementary set J_k.

The branching tree is constructed as follows. The subset of the first level is formed by fixing the assignment of the first server from I_k to different servers in J_k: X_{j1}, X_{j2}, $X_{j|Jk|}$.

Each set X_{j1} contains all variants in which the first failed server in J_k is assigned to server $j_1 \in J_k$, while the assignments of the remaining failed servers are arbitrary.

Similarly, the subset at the second level is formed by fixing the assignment of the second server in J_k to different servers in J_k. The set $X_{j1,j2}$ contains all variants in which the first failed server is assigned to server $j_1 \in J_k$, the second failed server is assigned to server $j_2 \in J_k$, and the assignments of the remaining servers in I_k remain arbitrary, and so on.

For each subset (i.e., each node of the branching tree), it is necessary to construct bounds of the objective function (12) and of the corresponding constraints. The general expression for the estimate of the objective function for the subset of variants $X_{j1,\ j2,\ jl}$, in this problem can be written as: $V(X_{j1,\ j2,\ ..,\ jl})$, where $V(X_{j1,\ j2,\ ..,\ jl})$ denotes the estimate of the objective function for all variants within the subset, with the first lll decision parameters fixed to j_1, j_2, .., j_l, while for the remaining parameters, $i = l + 1, l + 2, ..., |I_k|$, no specific assignment has yet been chosen.

This estimate is considered valid only if the following feasibility conditions for the constraints are satisfied:

$$\sum_{j=1}^{jk}\sum_{m=1}^{1} C_{mj} X_{mj} + min \sum_{m=1+1}^{Ik}\sum_{j=1}^{jk} X_m^{(k)} X_{mj}^{(k)} C_{mj} \le C$$

$$\sum_{j=1}^{jk}\sum_{m=1}^{1} V_j + min \sum_{m=1+1}^{Ik}\sum_{j=1}^{jk} V_i X_m^{(k)} X_{mj}^{(k)} < V_j^{max} - V_j, \qquad j=\overline{1, J_k}, \tag{16}$$

$$min\ \mu / \alpha_i^{(k)} > \lambda_0.$$

Based on the above formulas, the algorithm for solving the optimization problem is constructed as follows (Fig. 2):

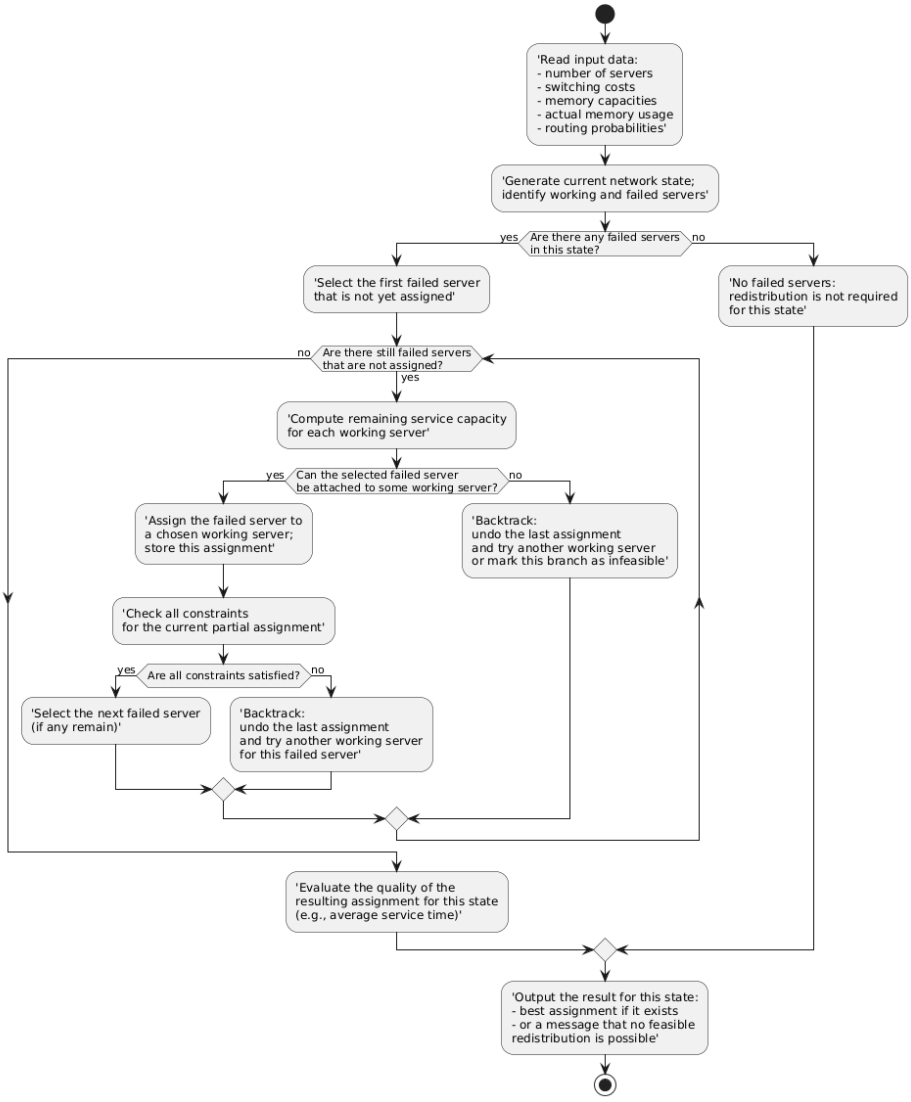

Figure 2: Algorithm for redistributing Users of failed servers.

Step 0. Initialization.

Step 1. Input data. Enter the initial parameters: the number of servers nnn; the switching costs of redirecting users from server S_i to server S_j, C_{ij} $i, j = 1, ..., n$; the maximum memory capacities of

the hardware hosting the servers, V_j^{max} j = 1, ..., n; the actual memory volumes of the servers, V_j j = 1, ..., n; the probabilities of request transmission from server S_i to server S_j, P_{ij} i, j = 1, ..., n; and compute the values P_{0i} i, j = 1, ..., n.

Step 2. Generation of the next state vector X_k. Determine the sets of operational and failed servers: I_0 (operational) and : I_1 (failed), respectively.

Step 3. Selection of the next unassigned failed server from I_1.

Step 4. Computation of the "residual service intensity". For each server in I_0, compute its residual service intensity, defined as the difference between the service intensity μ and the sum of request intensities from the server currently being considered and all failed servers already assigned to it. If the request intensity of the selected failed server in I_1 exceeds all residual service intensities of servers in I_0, proceed to Step 10. Otherwise, select the first server in I_0 whose residual service intensity exceeds the request intensity of the failed server. Record the pair (i, j), where $i \in I_1$ is the failed server and $j \in I_0$ is the selected operational server, into the assignment list.

Step 5. Constraint verification. Check whether constraints (11) and (12 are satisfied for the current partial assignment. If at least one constraint is violated, proceed to Step 10; otherwise continue.

Step 6. Check if the current failed server is the last element in I_1. If so, proceed to the next step; otherwise go to Step 10.

Step 7. Solving the system of equations. Solve system (11) using the simple iteration method to obtain α_i, j = 1, ..., n. Compute t_j, j = 1, ..., n, and the value T_k using formulas (7) and (10), respectively. Since the denominator in formula (12) satisfies the convergence condition, the iterative procedure converges.

Step 8. Verification of condition (12). If condition (12) is not satisfied, proceed to Step 10; otherwise continue.

Step 9. Update of the current best solution. If a previously computed value of T_{min} exists, compare it with the newly obtained value T_k. If $T_{min} < T_k$, continue to the next step. Otherwise, or if T_{min} has not yet been assigned, set $T_{min} = T_k$ and store the corresponding server assignment.

Step 10. Backtracking. Check whether backtracking is possible. Select the most recently assigned pair from the assignment list. If the list is empty, backtracking is impossible; proceed to Step 11. Otherwise, attempt to find another operational server to which the selected failed server can be reassigned. If such a server is found, update the assignment and go to Step 4. If no such server exists, remove the selected failed server from the assignment list and repeat Step 10.

Step 11. Output of results for the current state. If a value T_{min} has been obtained, output the corresponding optimal distribution of failed servers among operational servers. If no such value exists for the given state vector X_k, output a message stating that redistribution is impossible. If the number of processed states is less than 2^n - 2 increment the state index and return to Step 2; otherwise proceed to Step 12.

Step 12. Termination. The computational complexity $\theta_\alpha(n)$ of the proposed algorithm is significantly lower than the complexity $\theta(n)$ of the full enumeration method. As illustrated in Table 1, with an increasing number of servers, the efficiency of the algorithm — expressed as the ratio $\theta_\alpha(n)/\theta_\alpha(n)$ — grows, which confirms its practical advantage for authentication processes in terminal networks.

It should be noted that the inclusion of constraints (8)–(12) feasible optimal redistribution plan exists for assigning failed servers to the operational ones. In such cases, immediate operational measures must be applied to mitigate these situations. These measures may include relaxing the constraints by increasing the memory capacity of the relevant hardware components, replacing servers with more powerful units, or increasing the threshold value C in constraint (12).

The use of the proposed algorithm makes it possible to construct an optimal redistribution plan for assigning failed servers S_i to functioning servers S_j in the form of a matrix $\|X_{ij}(k)\|$ This matrix can serve as the basis for the decision-support functional block used by the security administrator of the terminal network of a university's corporate information system in emergency conditions.

Table 1
Computational complexity of the model

Number of terminal servers, n	Full enumeration complexity, $\theta(n)$	Proposed algorithm complexity, $\theta_\alpha(n)$	Efficiency ratio $\theta(n)/\theta_\alpha(n)$
3	9	6	1,5
4	40	24	1,7
5	195	80	2,4
6	1056	330	3,2
7	6321	1276	5,0
8	41392	5744	7,2
9	293607	21962	13,4
10	2237920	96910	23,0
11	18210092	668654	27,2

However, it should also be taken into account that a large number of Kerberos servers in the network increases the volume of authentication information exchanged between them, which, in turn, increases the overall load on the network.

4. Knowledge-Oriented Extension of the Model for Terminal Cluster Networks

In order to enhance the flexibility, scalability, and intelligence of the mathematical model of terminal cluster centers, it is advisable to integrate a knowledge-oriented approach whose central element is the construction of an ontology of terminal cluster systems and the mechanisms of its software interpretation. Unlike traditional models that operate exclusively with parametric descriptions, the ontology provides a structured, formalized, and semantically consistent representation of knowledge about the network, its behavior, constraints, and relationships between components. As a result, the mathematical model gains the ability to operate not only with flow intensities and cost values, but also with logical parameters, access policies, historical data, and contextual characteristics [18, 19].

The ontology of terminal cluster systems covers the fundamental elements of the infrastructure and their semantic links (Fig. 3). The central concept is the “Terminal Server” as an object characterized by a set of essential properties, including physical and logical resources, computational capacity, architectural type, failure probability, connections to other servers, and its role in authentication mechanisms. Each server is associated with the concept of “Memory Resource”, which specifies both the maximum available volume and the actual volume required to host authentication processes. The model also includes the concept of a “User Group”, which aggregates sets of users attached to specific servers and supports semantic labeling of different access profiles, priority levels, and request intensities. The state of each server is described by the concept of “Operational State”, which allows the system to capture normal functioning, partial degradation, or complete failure. All these elements are integrated into a structure that makes it possible to track the interactions between them, including routes of authentication information exchange, compatibility relations between servers, and dependencies between load and failure probability.

An important aspect of the ontological model is its ability to represent causal and semantic dependencies that are difficult to formalize within a purely analytical framework [20–23]. For example, a server with a high load level or frequent failures is automatically regarded as a less preferable candidate for load redistribution. Servers interconnected by high-speed communication channels receive higher priority for authentication processes, which reduces delays and increases

the overall throughput of the network. Security policies such as the mandatory use of a server with a higher trust level for certain categories of users can likewise be formalized within the ontology and automatically applied when generating redistribution plans.

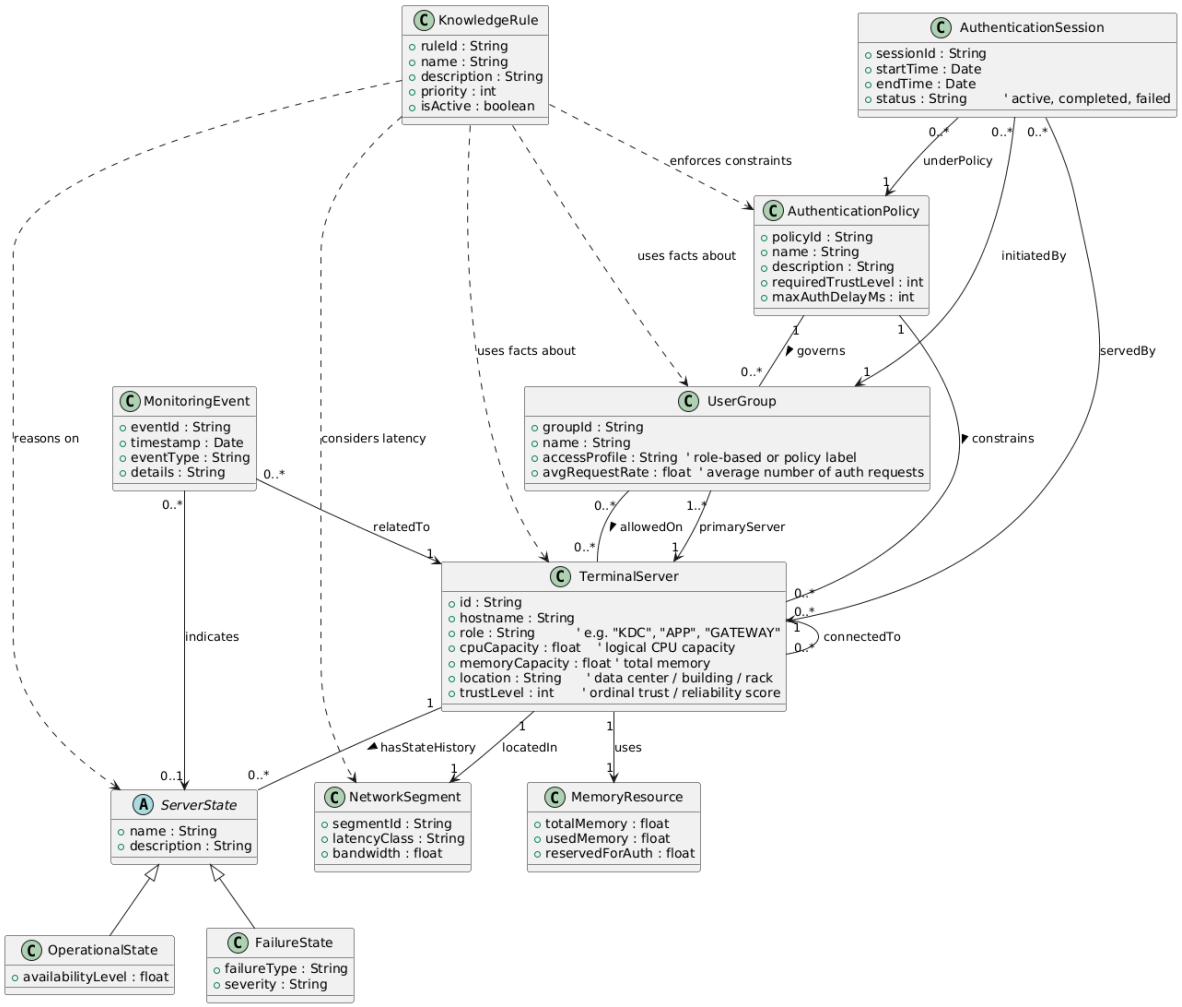

Figure 3: Ontology of terminal cluster system.

The software interpretation of such an ontology enables machine-level exploitation of the accumulated knowledge. By employing OWL and RDF standards, the ontology acquires a formal structure that can be processed in a software environment while preserving logical consistency and supporting automatic inference of new knowledge. Semantic queries expressed in SPARQL make it possible to extract complex dependencies, for instance, to determine the set of servers that simultaneously meet the requirements for throughput, latency, memory reserves, and historical reliability. As a consequence, the branches of the decision tree in the optimization algorithm are not explored exhaustively but are filtered according to semantic rules, which significantly reduces computational complexity.

Such an integration of semantic and mathematical layers makes it possible to construct solutions that are not only optimal in terms of formal criteria but also contextually appropriate and better aligned with the actual structure and behavior of the network. This enables a transition from reactive load redistribution in the event of failures to proactive management, allowing critical states to be predicted and the network configuration to be adapted based on continuously updated knowledge. Ultimately, the ontological extension of the model forms the foundation for an intelligent decision-support system that can explain its own conclusions, respond promptly to failures, and adapt to real operating conditions of terminal networks in higher education institutions.

5. Implementation and Experimental Research

The implementation of the proposed knowledge-oriented management system for terminal cluster centers is based on a modular architecture that integrates the ontological layer, the optimization core, the monitoring subsystem, and a web-based user interface (Fig. 4). This approach separates concerns across components, supports scalability, simplifies maintenance, and enables future extensions of the system without substantial modifications to the underlying codebase.

On the server side, the core of the system is the ontology management module, which operates on an OWL/RDF representation of the terminal cluster system model. The ontology is loaded into a semantic repository that supports SPARQL querying and logical reasoning, thereby providing access to up-to-date knowledge about terminal servers, user groups, operational states, authentication policies, and historical monitoring events. A service layer built on top of the ontology encapsulates the complexity of semantic operations and exposes a standardized programmatic interface to the remaining components. The optimization subsystem that implements dynamic load redistribution methods queries this service to obtain semantically filtered candidate servers, reduced state spaces, and constraints derived from knowledge and rules. The results of optimization — such as the optimal redistribution plan, the average service time, and the efficiency ratio $\theta(n)/\theta_\alpha(n)$ — are returned to the service layer and may be written back into the ontology as new facts or annotated decisions.

In parallel, the monitoring subsystem collects real-time data on the state of terminal servers, load levels, available memory resources, and failure events. These data are used both to update optimization model parameters and to enrich the ontology with new individuals of classes such as ServerState and MonitoringEvent. This enables the establishment of a feedback loop: monitoring produces events, the ontology accumulates structured knowledge, the optimization core makes decisions based on this knowledge, and the results of these decisions are fed back into the system for continuous refinement.

The user interface is implemented as a single-page web application with a modern adaptive design tailored for security administrators and network operations engineers (Fig. 5). The main view adopts a dashboard layout that aggregates key performance indicators, a detailed table of server states, an event and semantic decision log, and a panel with ontology concepts and SPARQL query examples. The page layout follows a two-column structure: a compact sidebar on the left presents summarized cluster information, while the right side contains the primary working area featuring server status tables and the chronological decision timeline.

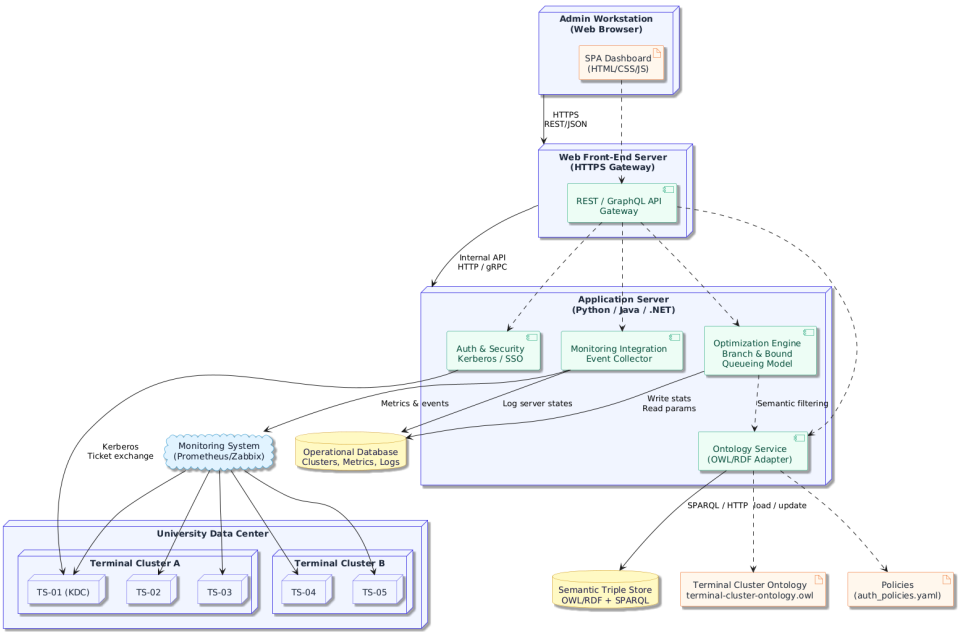

Figure 4: Deployment diagram of the ontology-driven terminal cluster management system.

At the top of the page, a header contains the system's branding, a concise textual description of the dashboard's purpose, and status indicators. The user receives immediate visual feedback on the activity of the monitoring subsystem and can initiate a redistribution process via an interactive control. A theme switcher (light/dark mode) is provided, implemented through dynamic CSS variable updates, to enhance usability under different lighting conditions.

The sidebar includes an overview block displaying aggregated indicators: the number of terminal servers, the number of currently active nodes, the current average authentication service time T, and the integrated efficiency metric $\theta(n)/\theta_{\alpha}(n)$, which quantifies the improvement obtained by combining branch-and-bound optimization with ontological filtering. A compact list of core ontology concepts is presented as a visual legend to support interpretation of the semantic layer. Below, a SPARQL query fragment illustrates how the system selects candidate servers according to trust values, available memory, and policy constraints.

The main area contains a detailed table of terminal server states with information on cluster membership, operational status (online, degraded, failed), load levels, trust ratings, and available memory. Visual markers — such as colored badges and status pills—help administrators quickly identify critical nodes and evaluate load distribution. Adjacent to the table is a chronological decision log that records detected failures, results of semantic candidate selection, key steps in the optimization workflow, and the final redistribution plan. Each entry provides a brief event description and contextual explanation of which ontological elements or KnowledgeRule constraints were involved in the corresponding decision.

The page concludes with an explanatory section summarizing how the knowledge-oriented decision was produced: from failure detection and semantic reasoning, through candidate filtering and constraint enforcement, to the final optimization step operating on a reduced search space. This section serves as an element of explainable analytics, essential for integrating decision-support components into critical infrastructure.

Overall, the system implementation integrates semantic technologies, optimization methods, and a modern web interface to support the full operational cycle: monitoring → ontological modeling → optimization computation → visualization and explanation of decisions. This approach increases the resilience and efficiency of authentication channels in terminal cluster centers of university networks while ensuring transparency, interpretability, and usability for expert administrators.

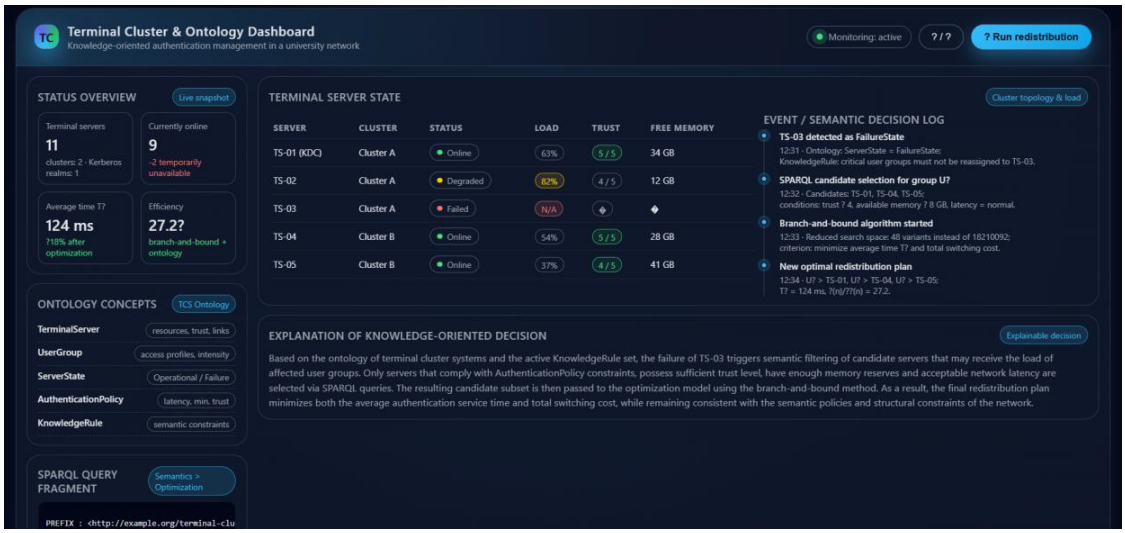

Figure 5: System of the knowledge-based terminal server load management system.

6. Conclusion

This study develops a comprehensive knowledge-oriented approach aimed at enhancing the resilience, efficiency, and intelligence of terminal cluster systems responsible for user

authentication in the corporate networks of higher education institutions. By integrating mathematical modeling, semantic technologies, and advanced optimization methods, the research presents a unified framework for the dynamic redistribution of load among terminal servers under conditions of failures, peak loads, and limited communication bandwidth.

One of the key outcomes of the study is the construction of a formal model of terminal cluster centers represented as an open stochastic network. This model enables accurate estimation of flow intensities, authentication service time, and the impact of server failures on the global performance indicator T_0. The research demonstrates that the problem of optimal redistribution of user groups among operational servers belongs to the class of high-complexity combinatorial problems, while exhaustive enumeration is impractical due to the exponential growth of candidate configurations.

A central innovation of this work is the development of an ontology of terminal cluster systems, which enables semantic representation of the network structure, server resources, access policies, logical dependencies, historical states, and behavior patterns. The ontology formalizes relationships among system components, ensures the logical consistency of knowledge, and enables automated inference. Its integration with the mathematical model significantly improves the relevance and correctness of redistribution decisions, since semantic restrictions automatically eliminate infeasible, conflicting, or suboptimal variants before numerical optimization begins.

The proposed algorithm, which combines a branch-and-bound method with semantic filtering based on SPARQL queries and KnowledgeRule constraints, substantially reduces computational complexity. Experimental results confirm that semantic technologies reduce the search space by orders of magnitude, enabling rapid construction of optimal redistribution plans and achieving a significant reduction of the average authentication service time T_0. At the same time, the approach ensures compliance with technical, policy, and contextual constraints—an outcome unattainable in purely numerical models.

A full-scale software system was implemented to validate the proposed approach. It includes an OWL/RDF knowledge repository, a SPARQL query engine, an optimization core, a real-time server monitoring subsystem, and a modern web interface. The developed dashboard provides intuitive visualization of cluster parameters, server state tables, event and decision timelines, and semantic explanations of the reasoning process. This architecture enables prompt reaction to failures, enhances the transparency of system behavior, and increases administrator trust in automated decision-making.

The results of the study demonstrate that combining mathematical optimization with ontology-based knowledge representation forms a solid foundation for the development of intelligent management systems for university-scale network infrastructures. The proposed approach improves the availability of critical services, minimizes authentication delays, reduces the impact of failures on end users, and ensures adaptive behavior of network infrastructure under dynamic conditions and increasing workload.

Promising directions for future research include integrating predictive failure models based on machine learning, extending the ontology to support multi-realm authentication architectures, applying causal analysis methods to assess the impact of configuration changes, and incorporating Explainable AI techniques to improve transparency and interpretability of system recommendations. Collectively, these developments may lead to a new class of decision-support systems for managing distributed infrastructures with high requirements for reliability and security.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly to check grammar and spelling, paraphrase, and reword the text. These tools help identify and correct grammatical errors, typos, and other writing mistakes, improving the clarity and professionalism of the text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Wang, J. Li, X. Cao, C. Lv, L. Yang, Research on computing power resources-based clustering methods for edge computing terminals, Appl. Sci. 15 (20) (2025). doi:10.3390/app152011285.

[2] X. Cao, C. Chen, S. Li, C. Lv, J. Li, J. Wang, Research on computing task scheduling method for distributed heterogeneous parallel systems, Sci. Rep. 15 (2025). doi:10.1038/s41598-025-94068-0.

[3] W. Zhou, H. Guo, L. Yao, Statistical modeling of traffic flow in commercial clusters based on a street network, Sustainability 15 (3) (2023). doi:10.3390/su15031832.

[4] S. Ayaz, K. S. Khattak, N. Z. H. Khan, Minallah, M. A. Khan, A. N. Khan, Sensing technologies for traffic flow characterization: From heterogeneous traffic perspective, J. Appl. Eng. Sci. 20 (1) (2022) 29–40. doi:10.5937/jaes0-32627.

[5] A. Benghalia, A. Ferdjallah, M. Oudani, J. Boukachour, Machine learning and simulation for efficiency and sustainability in container terminals, Sustainability 17 (7) (2025). doi:10.3390/su17072927.

[6] S. El Mekkaoui, L. Benabbou, A. Berrado, Machine learning models for efficient port terminal operations: Case of vessels' arrival times prediction. IFAC-PapersOnLine 55 (10) (2022) 3172–3177. doi:10.1016/j.ifacol.2022.10.217.

[7] O. Sherstiuk, O. Kolesnikov, V. Gogunskii, K. Kolesnikova, Developing the adaptive knowledge management in context of engineering company project activities, Int. J. Comput. 19 (4) (2020) 590–598. doi:10.47839/ijc.19.4.1993.

[8] S. Maslovskyi, A. Sachenko, Adaptive test system of student knowledge based on neural networks, in: Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS '2015, IEEE, New York, NY, 2015, pp. 940–944. doi:10.1109/IDAACS.2015.7341442.

[9] D. Drungilas, M. Kurmis, A. Senulis, Z. Lukosius, A. Andziulis, J. Januteniene, M. Bogdevicius, V. Jankunas, M. Voznak, Deep reinforcement learning based optimization of automated guided vehicle time and energy consumption in a container terminal, Alex. Eng. J. 67 (2023) 397–407. doi:10.1016/j.aej.2022.12.057.

[10] R. Shevchuk, V. Martsenyuk, B. Adamyk, V. Benson, A. Melnyk, Anomaly detection in blockchain: A systematic review of trends, challenges, and future directions, Appl. Sci. 15 (15) (2025). doi:10.3390/app15158330.

[11] Q. Zeng, Z. Yang, X. Hu, A method integrating simulation and reinforcement learning for operation scheduling in container terminals, Transport 26 (2011) 383–393. doi:10.3846/16484142.2011.638022.

[12] R. Choe, J. Kim, K. R. Ryu, Online preference learning for adaptive dispatching of AGVs in an automated container terminal, Appl. Soft. Comput. 38 (2016) 647–660. doi:10.1016/j.asoc.2015.09.027.

[13] A. Dávila de León, E. Lalla-Ruiz, B. Melián-Batista, J. M. Moreno-Vega, A machine learning-based system for berth scheduling at bulk terminals. Expert Syst. Appl. 87 (2017) 170–182. doi:10.1016/j.eswa.2017.06.010.

[14] Z. Wang, H. Chen, H. Tang, L. Zheng, J. Zheng, Z. Liu, Z. Hu, A three-layer coordinated planning model for source-grid-load-storage considering electricity-carbon coupling and flexibility supply-demand balance, Sustainability 17 (16) (2025). doi:10.3390/su17167290.

[15] P. Torres-Bermeo, K. López-Eugenio, C. Del-Valle-Soto, G. Palacios-Navarro, J. Varela-Aldás, Sizing and characterization of load curves of distribution transformers using clustering and predictive machine learning models, Energies 18 (7) (2025). doi:10.3390/en18071832.

[16] F. Zhou, C. Wu, Y. Wang, Q. Ye, Z. Tai, H. Zhou, Q. Sun, Collaborative optimization of cloud-edge-terminal distribution networks combined with intelligent integration under the new energy situation, Mathematics 13 (18) (2025). doi:10.3390/math13182924.

[17] P. Jiang, X. Dou, J. Dong, H. Huang, Y. Wang, Terminal node of active distribution network correlation compactness model and application based on complex network topology graph, Sustainability 15 (1) (2023.). doi:10.3390/su15010595.

[18] M. Dyvak, A. Kovbasistyi, A. Melnyk, I. Shcherbiak, O. Huhul, Recognition of relevance of web resource content based on analysis of semantic components, in: Proceedings of the 9th International Conference on Advanced Computer Information Technologies, ACIT ’2019, IEEE, New York, NY, 2019, pp. 297–302, doi:10.1109/ACITT.2019.8779897.

[19] A. Bomba, M. Nazaruk, N. Kunanets, V. Pasichnyk, Constructing the diffusion-like model of bicomponent knowledge potential distribution, Int. J. Comput. 16 (2) (2017) 74–81. doi:10.47839/ijc.16.2.883.

[20] T. Lendyuk, O. Bodnar, S. Rippa, A. Sachenko, Ontology application in context of mastering the knowledge for students, in: Proceedings of the XIII International Scientific and Technical Conference on Computer Science and Information Technologies, CSIT ’2018, IEEE, New York, NY, 2018, pp. 123–126. doi:10.1109/STC-CSIT.2018.8526710.

[21] R. E. Hiromoto, Parallelism and complexity of a small-world network model, Int. J. Comput. 15 (2) (2016) 72–83. doi:10.47839/ijc.15.2.840.

[22] O. Androshchuk, R. Berezenskyi, O. Lemeshko, A. Melnyk, O. Huhul, Model of explicit knowledge management in organizational and technical systems, Int. J. Comput. 20 (2) (2021) 228–236. doi:10.47839/ijc.20.2.2170.

[23] E. Oikonomou, A. Rouskas. Efficient schemes for optimizing load balancing and communication cost in edge computing networks, Information 15 (11) (2024). doi:10.3390/info15110670.