DOI: https://doi.org/10.15276/aait.09.2026.23
UDC 004.75

Model for storing structured data in heterogeneous distributed programmable mobile systems

Oleksandr O. Fomin[1)]
ORCID: https://orcid.org/0000-0002-8816-0652; fomin@op.edu.ua. Scopus Author ID: 57103429400
Andriy M. Chmelevskyi[1)]
ORCID: https://orcid.org/0009-0008-6450-6875; stech@stud.op.edu.ua
[1)] Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

The rapid development of mobile ad hoc networks and edge computing paradigms necessitates the design of resilient storage architectures capable of operating efficiently under stochastic topology changes. Existing decentralized storage approaches exhibit prohibitive signaling overhead and integrity degradation during critical link failures, making adaptive information management **highly relevant**. **The purpose of the article** is to ensure high availability and integrity of information in distributed programmable mobile systems in environments with dynamic network topologies and limited resources of mobile nodes. To achieve this, **the following tasks were formulated**: to propose a hybrid structured data storage model based on semantic standards; to develop an adaptive replication algorithm predicting device trajectories; and to experimentally evaluate the effectiveness of the developed solutions. **The following methods** were applied: graph theory for network topology modeling, the software-defined networking and software-defined storage paradigms for resource orchestration, and mathematical modeling techniques for calculating link expiration time. **The scientific novelty** of the work lies in the improvement of the structured data storage model through the integration of software-defined network paradigms and heuristic adaptive replication with competitive aging mechanisms, resolving the severe compromise between consistency and availability. **The practical significance** of the obtained results lies in their direct applicability to intelligent transportation systems, smart cities, and emergency response networks during disaster recovery. **The most significant results** demonstrate that the proposed model maintains a request delivery success rate of eighty-three point nine percent at movement speeds up to one hundred kilometers per hour while simultaneously reducing the average data lookup delay from two hundred and seven milliseconds to eighty-seven milliseconds. **Conclusions**. Developing a hybrid model based on intelligent orchestration and a two-level consistency policy successfully enables bounded inconsistency during network partitioning, enabling continuous application operability within isolated segments with a guaranteed delayed restoration of global semantic data integrity.

Keywords: Modeling; mobile ad-hoc networks; heterogeneous systems; structured data; data replication; dynamic topology; edge and fog computing; data consistency

***For citation*: Fomin O. O., Chmelevskyi A. M. "Model for storing structured data in heterogeneous distributed programmable mobile systems". *Applied Aspects of Information Technology.* 2026; Vol.9 No.3: 340–352. DOI: https://doi.org/10.15276/aait.09.2026.23**

INTRODUCTION

The current stage of information technology development is characterized by rapid growth in the number of mobile programmable devices that are integrated into dynamic distributed systems, such as Mobile Ad-hoc Networks (MANET) [1], [2] or Fog computing [3], [4], [5]. The evolution of distributed systems and the widespread deployment of intelligent devices at the network's edge have driven the transition from centralized data processing architectures to decentralized computing paradigms. This approach is successfully used in practical applications of autonomous transport (real-time storage and exchange of maps of the local environment), the Industrial Internet of Things (storage and provision of access to data on the location and status of distributed resources), etc. A key feature orchestration and adaptive data processing at the network edge.

© Fomin O., Chmelevskyi A., 2026

of such systems is the ability of nodes not only to transmit but also to autonomously process and store structured data (data with a clearly defined schema and semantic relationships) [2], [4].

In real-world heterogeneous networks, where nodes differ in hardware, connection quality, and data distribution, the task of ensuring the availability and integrity of structured data managed by distributed systems becomes critical [6].

Thus, storing structured data in heterogeneous networks faces a critical contradiction: the need to ensure the integrity and high availability of information on the one hand, and the severe resource constraints of mobile nodes (power consumption, memory capacity, and communication channel bandwidth) on the other, under conditions of constantly changing topology and unstable communication channels in a distributed system. This work aims to develop a hybrid model for storing structured data that resolves this contradiction through intelligent resource.

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

1. LITERATURE REVIEW AND PROBLEM FORMULATION

The development of data storage models in distributed systems has evolved from centralized relational databases to flexible decentralized solutions based on edge computing [7], [8]. In this work, a distributed data store is defined as a structure in which information is stored across multiple physical or virtual nodes, acting as a single logical mechanism to ensure high availability, fault tolerance, and performance. Structured data, characterized by a clear model (e.g., data in relational DBMSs), requires complex synchronization algorithms in a distributed environment to support ACID properties (atomicity, consistency, isolation, durability).

One of the most fundamental models is the Hadoop Distributed File System (HDFS), which is optimized for processing large volumes of data through horizontal scaling and high fault tolerance. HDFS uses a "master-slave" architecture, where the NameNode manages metadata, and DataNodes store the data blocks directly. Replication in such systems is typically static: data is copied to multiple nodes (three copies by default) to ensure system survivability in the event of individual component failures [9].

Alongside file systems, distributed NoSQL systems such as Apache Cassandra have evolved, utilizing a decentralized ring topology with no single point of failure. Cassandra is based on an architecture where every node is equal, and data consistency is achieved through Gossip protocols. For structured data, NoSQL systems offer flexible schemas that allow for the storage of large volumes of information with low read and write latency, which is crucial for web services and real-time analytics [10], [11].

Recently, cloud services have also been actively using object storage, such as Amazon S3, and blockchain systems (e.g., Hyperledger Fabric) to ensure the integrity of structured data in distributed systems [12], [13]. A comparison of the discussed technologies for storing structured data in distributed systems is presented in Table 1.

Models based on Software-Defined Networking (SDN) marked an important milestone in this development. SDN allows the control plane to be separated from the data plane, opening up possibilities for programmable control of data flows and their routing [14]. In the context of data storage, SDN enables the virtualization of storage resources, transforming hardware into flexible software resource pools that can be dynamically allocated depending on the load [15].

Table 1. **Comparison of modern technologies for storing structured data in distributed systems**

Characte-ristic	Relational systems (RDBMS)	NoSQL (Cassandra, MongoDB)	Object storage (S3)	Block-chain storage
Data structure	Tables with a fixed schema	Collections, key-value pairs	Objects with metadata	Blockchains (hashes)
Consistency	Strict (ACID)	BASE	Bounded/ Strict	Full consensus
Scalability	Predomi-nantly vertical	Horizontal (high)	Virtually unlimited	Limited
Fault tolerance	Master-slave replication	Decentra-lized ring	Geo-replication	Multi-node redundancy
Main advantage	Transaction integrity	Speed and flexibility	Low storage cost	Immutabi-lity and security

***Source*: compiled by the authors**

Shortcomings of existing data storage models in distributed systems with dynamic mobile topologies. Despite the significant number of existing models, most were designed for relatively stable data centers, making them ineffective in dynamic mobile topologies. The main problem with the considered data storage models is their failure to account for the constant connection and disconnection of nodes, limited energy resources, and the variability of communication channels in mobile networks [2], [3], [15].

The first critical drawback in classical distributed systems is centralized management. Thus, in single-leader models, the failure of this leader in a mobile environment triggers a lengthy re-election process, during which the system remains unavailable for writing.

The second drawback is the static nature of replication mechanisms. Existing models (Cassandra, HDFS, etc.) typically use a fixed number of copies distributed across nodes selected at the time of write. In mobile systems, where nodes constantly change their geographic and network location, static replicas quickly end up "far" from active users, leading to increased latency and inefficient use of network bandwidth. Such systems lack mechanisms for predicting mobility, so they cannot proactively move data to the nodes where the user is heading.

The third drawback is the high computational complexity of these solutions. For instance, blockchain-based technologies require each node to store a full copy of the ledger and participate in a

resource-intensive validation process. For mobile devices with limited memory and battery life, this is unacceptable. In addition, blockchain systems exhibit high write latency, making them unsuitable for processing large streams of structured data in real time.

The fourth drawback is the limitations of traditional SDN in highly mobile environments. Although SDN allows for dynamic changes to network configuration, a centralized controller often cannot keep up with rapid topology changes in mobile networks, leading to packet loss and delays in data routing. Furthermore, existing SDN solutions for data storage typically focus on managing traffic flows rather than on the intelligent management of the lifecycle of structured data replicas.

Table 2 presents a classification of the identified shortcomings in structured data storage within existing distributed systems.

Table 2. **Classification of shortcomings in structured data storage in existing distributed systems**

Category of shortcomings	Specific manifestation	Consequences for mobile systems
Algorithmic	Dependence on a single leader (Raft/Paxos)	Temporary unavailability during topology changes
Architectural	Static placement of replicas	High latency during node movement
Network	High overhead for consensus	Overloading of narrow mobile communication channels
Resource	Memory and energy requirements (Blockchain)	Rapid degradation of device battery life
Security	Complexity of rights management in MANET	Risk of unauthorized access during handover

***Source*: compiled by the authors**

Ultimately, existing models fail to provide an adequate balance between data integrity and availability during network partitioning. According to the CAP theorem, in the event of a communication failure, the system must choose either integrity or availability [16], [17]. Most industrial systems (e.g., ETCD, MongoDB) prioritize integrity, which in mobile environments leads to frequent service failures when a device is outside the area of stable coverage.

Statement of the research problem

The problem lies in resolving the conflict between the need to maintain high availability A and integrity I of structured data D_{struct} in a dynamic topology $T(t)$ and the constraints on nodes regarding energy E_{lim}, memory $M_{lim,}$ and bandwidth B_{lim}. Formally, the optimization problem can be formulated as the minimization of the cost functional while ensuring service quality:

$$\min J = \int_0^T \left[\alpha E(t) + \beta L(t)\right] dt, \quad A \geq A_{min},\ I \geq I_{min},\ E \geq E_{lim},\ M \geq M_{lim}, \tag{1}$$

where α, β are weighting coefficients, $E(t)$ is instantaneous power consumption, and $L(t)$ is data access latency.

2. RESEARCH OBJECTIVES AND TASKS

The objective of the research is to ensure high availability and integrity of information in distributed programmable mobile systems in environments with dynamic network topologies and limited resources of mobile nodes by developing a mathematical model and an algorithm for dynamic replication and verification.

To achieve this objective, *the following tasks* are set in this work:

1. Propose a hybrid model for storing structured data, adapted for fragmentation in mobile edge systems, based on the JSON-LD and NGSI-LD semantic standards.
2. Develop an adaptive replication algorithm that uses node trajectory prediction and the state of their resources to select the optimal number and location of data copies.
3. Justify the performance criteria and conduct an experimental study of the proposed solutions in comparison with existing approaches in terms of availability, integrity, and energy efficiency.

3. HYBRID MODEL FOR STRUCTURED DATA STORAGE

3.1. Structure of the Hybrid Model

To overcome the limitations of the considered structured data storage models, the inefficiency of static replication, and consistency issues in MANETs, a hybrid model is proposed. This model is based on the integration of intelligent network management and a decentralized data verification mechanism. The model is organized to bring data as close as possible to the consumer while maintaining global integrity through distributed control. The model's architecture consists of three hierarchical levels:

1. *Physical layer (mobile node layer).* This layer represents a dynamic graph of heterogeneous devices (ranging from ordinary smartphones and IoT sensors to unmanned aerial vehicles and VANETs), which unites physical devices into a single network. Each node allocates a memory quota to a shared virtualized space. Thanks to the SDS paradigm, all disparate and incompatible memory formats are abstracted by the operating system and unified into a single scalable logical space independent of hardware. At this level, nodes function as passive storage devices and switches, responsible solely for data caching, energy-efficient packet forwarding, and executing direct commands from the management layer.

A graph model based on the NGSI-LD standard, which uses JSON-LD as a serialization format, was chosen for the mathematical description of this level [18], [19]. Each data element (entity) is represented as a node V of the graph, and the relationships between them as edges E. Entity attributes and their metadata are modeled as node properties. Mathematically, the data structure is described as a multigraph $G_1(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{P}, \boldsymbol{R})$, where: $\boldsymbol{V}$, $\boldsymbol{E}$, $\boldsymbol{P}$, $\boldsymbol{R}$ are the sets of entities, relationships between entities, properties, and rules for adding metadata to relationships and properties, respectively.

2. *Logical management and orchestration layer.* This layer separates decision-making logic from physical data transmission and is implemented through a set of dynamically selected SDN controller nodes. In a decentralized environment, these may be the most powerful and stable devices (selected via leadership algorithms), or, in the presence of edge infrastructure, dedicated controller clusters that ensure the absence of a single point of failure. The controllers maintain a global virtual network map in RAM, aggregating a continuous stream of telemetry from all nodes: precise coordinates, velocity vectors, battery charge levels, and channel quality (RSSI, SNR). AI-based algorithms support decisions regarding preventive routing and context-based topology failure prediction, determining where a replica should be migrated, when to initiate its replication, and how to route user requests while bypassing interference zones.

3. *Consistency Management Layer.* Responsible for ensuring that stored data remains correct and secure even when the network suffers from physical disruptions. Since the network is prone to physical disruptions, traditional consensus protocols (such as Paxos or Raft) would lead to a complete collapse. Therefore, a two-level consistency model is implemented at this layer. The entire structured data set is divided into two logical clusters based on the semantic proximity of the information or the geographical location of the consumers.

– *Intra-cluster behavior.* Within a stable physical cluster (where connectivity is reliable and latency is minimal), the controller applies strong consistency protocols. All write operations are guaranteed to be committed simultaneously, ensuring perfect consistency.

– *Inter-cluster behavior.* Between spatially isolated domains (when the network breaks down into isolated islands), the system allows for a mode of bounded or controlled inconsistency.

3.2. Transaction Interface and Disconnection Compensation

To implement the aforementioned consistency, the model provides applications with a dual-database interface, distinguishing between “strong” and “weak” transactions. If a mobile user moves away and loses connection with the network core, they continue to operate in offline mode, using “weak” read and write operations. These interact exclusively with local replicas, which may potentially be out of date. Changes made by a “weak” write are recorded in the local log as provisionally accepted. As soon as the SDN controller detects that connectivity has been restored, it automatically initiates a background data reconciliation process: local logs are merged, conflicts are resolved based on vector clocks or semantic business rules, and transactions are transitioned to “strong” status. This resolves the conflict between availability and integrity: applications never lock up, but the system eventually restores strict integrity.

Additional services may also operate at the consistency management level, such as a security and audit module to ensure basic security objectives, such as protection against data manipulation, ensuring location confidentiality, authentication, etc.

A block diagram of the proposed model for storing structured data in distributed programmable mobile systems is shown in Fig. 1.

The central component of the proposed model for storing structured data is the logical management and orchestration layer, which is responsible for distributing replicas among the nodes of distributed programmable mobile systems. To ensure the effective operation of the proposed model and guarantee high availability and data integrity in distributed programmable mobile systems under conditions of dynamic network topology and limited

mobile node resources, an algorithm for dynamic replication and verification based on stable neighbors with a competitive aging mechanism has been developed. The algorithm must continuously modify the spatial distribution scheme of objects, maximizing the probability of their presence within the nearest reach and minimizing transmission costs and the risk of loss due to disconnections.

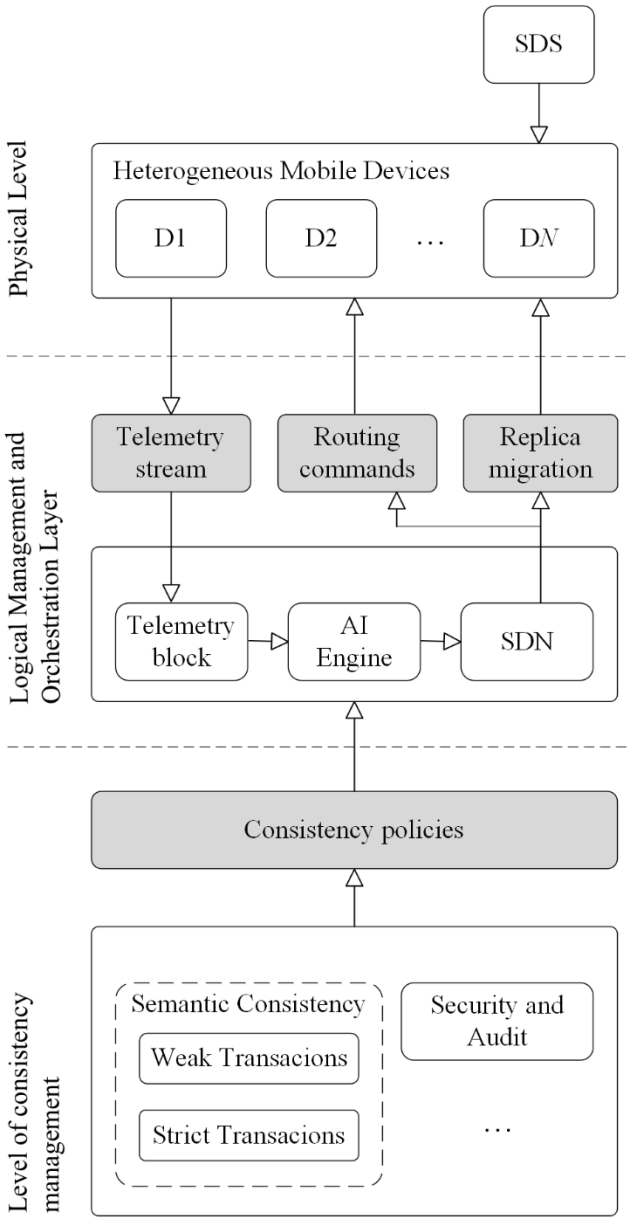

Fig. 1. **Block diagram of a model for storing structured data in distributed programmable mobile systems**
Source: **compiled by the authors**

To formalize the optimization problem solved by the algorithm, the MANET network is represented as an undirected dynamic graph $G_2(\boldsymbol{V}, \boldsymbol{E}, t)$, where *t is* a time step. For each node $i \in \boldsymbol{V}$, the following critical resource parameters are defined: current remaining energy $B_i(t)$, maximum battery capacity B_{max}, discharge rate λ, and the amount of memory allocated for shared use C_i. The repository operates on a set of structured objects (or database blocks) $\boldsymbol{O} = \{o_1, o_2, \ldots, o_m\}$. Each object o_k is characterized by its replication scheme $R(o_k) \subset \boldsymbol{V}$, which is the set of nodes currently containing a copy of it.

Unlike static systems, the developed algorithm continuously evaluates data demand. The SDN controller collects the metric $AC_{i,k}$, which denotes the absolute number of requests (for reading and writing) to object o_k initiated by node *i* over a given time interval Δt.

3.3. Calculation of access metrics

However, simply summing the requests may result in placing a replica on a node with a critically low charge level, causing its failure. Therefore, the algorithm introduces the concept of the weighted number of read-write requests (WAC), penalizing nodes with low charge:

$$WAC_{i,k} = AC_{i,k} \, (B_i(t) / B_{max})^{\alpha}, \qquad (2)$$

where α is the configuration coefficient. The higher the current charge $B_i(t)$, the greater the weight of its requests, which prompts the algorithm to place replicas closer to energetically stable nodes.

3.4. Topology Stability Metric

High mobility of network nodes negates the benefit of replication if a replica migrates to a node that will leave the network shortly. Therefore, the controller must calculate the Link Expiration Time (LET) metric. This metric is calculated based on the coordinates and relative vector velocity of two nodes.

Let two mobile nodes *i* and *j* have the same wireless transmission radius *r*. At time $t_0 = 0$, their spatial coordinates are (x_i, y_i) and (x_j, y_j), respectively. The nodes move at speeds v_i and v_j at angles (directions) θ_i and θ_j. The projections of the velocities (vector components) onto the coordinate axes are calculated as: $v_{xi} = v_i \cos(\theta_i)$, $v_{yi} = v_i \sin(\theta_i)$, $v_{xj} = v_j \cos(\theta_j)$, $v_{yj} = v_j \sin(\theta_{(j)})$.

To simplify the final formula, auxiliary variables are introduced that represent the differences in velocities and coordinates:

$$a = v_{xi} - v_{xj}, \; b = x_i - x_j, \; c = v_{yi} - v_{yj}, \; d = y_i - y_j. \qquad (3)$$

Based on this data, the *LET* time during which the two nodes will remain within range of each other (until the connection is lost) is determined by solving the quadratic equation of the form:

$$LET = \frac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2}. \quad (4)$$

Routing algorithms and SDN controllers use this value to predict connection stability: the higher the *LET* value, the longer the nodes will move along compatible vectors (for example, in the same direction at similar speeds), and the more reliable this link is for placing or migrating data replicas.

Global objective function. Accessing a replica incurs a certain communication cost. The cost $Cost(i, o_k)$ is determined by the number of logical or physical hops $h(i, v)$ to the nearest node $v \in R(o_k)$, multiplied by the energy cost of transmission E_{tx}. The goal of the algorithm is to minimize the total cost function F for the entire system under strict constraints (the total volume of replicas at a node cannot exceed its quota C_i, and the energy of the carrier node must not fall below the threshold value B_{crit}):

$$\min F = \sum_{k=1}^{m}\sum_{i\in V}\left[WAC_{i,k} \cdot \min_{v\in R(o_k)} (h(i,v)\cdot E_{tx}) \right], \quad (5)$$

where m *is* the total number of unique structured objects (or data blocks) in the repository; $R(o_k)$ is the set of nodes on which copies (replicas) of object $o_{(k)}$ are currently located; $h(i, v)$ is a communication distance: the number of network hops between node i and node v.

Criterion (5) means that the system sums the access cost for all queries to all objects. The task of the optimization algorithm is to find a spatial placement scheme for the replicas $R(o_k)$ such that the data is located as close as possible to the nodes that request it most frequently and that have sufficient energy to process it.

3.5. Dynamic Replication and Verification Algorithm

The optimization process for the proposed hybrid model is performed iteratively and consists of the following steps:

Step 1. Telemetry monitoring and aggregation. In this step, the controller collects responses from all nodes and constructs a matrix containing the coordinates (x_i, y_i), the current residual energy $B_i(t)$, free memory, and the matrix $AC_{i,k}$.

Step 2. Spatial clustering. Based on the link stability factor $S(i,j)$ (calculated based on relative vector velocity, received signal level, encounter history, and continuous time spent in line of sight), the controller identifies stable subgraphs (clusters) within the graph. Nodes that demonstrate a high community coefficient (e.g., a group of vehicles moving in the same direction on a highway) are grouped into trusted groups.

Step 3. Localization of "hot" data. Access logs are processed to identify "hot objects." If the weighted aggregate demand *(WAC)* for a specific set of data in a given cluster exceeds the set threshold θ_{hot}, the controller decides on the need for intervention: the existing replica must either be replicated or migrated closer to the center of demand.

Step 4. Adaptive data placement. For the selected object, the controller sorts all members of the stable cluster by a comprehensive suitability criterion that takes into account available free space, high battery charge, and central position within the group. The controller commands the data to be copied to the optimal node. If the current replica carrier signals imminent power loss or a change in motion vector (moving away from the group), a *proactive migration* process is initiated. The replica is smoothly transferred to another, more stable node, even before the physical connection is severed.

Step 5. Data aging assessment. In cases where the candidate node's memory is full, a replacement algorithm is applied. Unlike the primitive replacement of the oldest files, which creates excessive load on the storage devices, the use of a competitive aging mechanism is proposed. This mechanism takes into account the frequency of access and long-term access history, which allows for a reduction in the number of physical write operations to disk compared to conventional greedy strategies [20], thereby conserving the hardware resources of mobile platforms.

For each data object o_k stored on mobile node i, a dynamic freshness coefficient $V_i(o_k, t)$ is introduced, which is calculated using the following formula:

$$V_i(o_k, t) = V_i(o_k, t_{last}) e^{-\gamma(t-tlast\}} + ln(f_i(o_k) + 1), \quad (6)$$

where t_{last} is the time of the last access to object o_k; $\gamma > 0$ is the data decay (aging) coefficient; $f_i(o_k)$ is the frequency of requests to the object during the last interval Δt.

When a node's free memory reaches a critical threshold ($C \le C_c$), a competitive replacement procedure is initiated. The node removes the object with the minimum current value of the relevance coefficient $V_i(o_k, t)$. This prevents premature updating of relevant data that has not been temporarily requested due to short-term communication interruptions [21]. Delaying the removal of modified blocks and reusing them in the cache avoids erase/write cycles in flash memory,

which mathematically justifies a reduction in the number of write operations compared to classical algorithms that operate "blindly" with respect to the type of operations.

Step 6. Integrity Check During Access. When directly handling user requests, the controller routes packets not simply along the shortest path, but along the most reliable path calculated by optimization algorithms, avoiding congested areas. Upon receiving data, an audit procedure is initiated. The auditor or the user themselves generates a verification equation based on aggregated tags:

$$T - RES = (r_2 - r_1)M, \quad (7)$$

where T is the calculated check value, which is generated independently by an independent auditor based on stored tags; *RES* is the response returned by the peripheral node (replica carrier) to the auditor's request; r_1 and r_2 are random numbers or secret masking parameters; M is the block of structured data whose integrity is being verified.

The basic prototype for expression (7) is an interactive zero-knowledge proof (ZKP) scheme based on a modified Schnorr protocol for distributed content storage systems. The parameters r_1 and r_2 are random secret masking factors that are generated locally on the end-user's mobile device and are not transmitted over the network in plaintext. Computing the original values of r_1 and r_2 is impossible due to the complexity of the discrete logarithm problem in a finite field. The integrity tag T itself, which is transmitted to the auditor, is protected by an authentication tag. Any modification of the T tag or data blocks by an attacker during transit leads to a violation of the identity in the verification formula, which is immediately detected by the auditor without downloading the file itself.

If mathematical equality (7) holds (*True*), the object is considered intact and uncompromised. Otherwise, the system marks the node as malicious and blocks access to its replicas.

The logic of the audit process is as follows: an independent auditor compares the *RES* proof sent by the carrier node with its own pre-generated value of T. If the difference between them is exactly equal to the product of the difference between the masking numbers and the data block, i.e., $(r_2 - r_1)\ M$, this mathematically proves that the node indeed stores the original, unaltered data, and the system returns a successful verification result. If the data has been corrupted, the mathematical equality will not hold.

4. ORGANIZATION OF THE EXPERIMENT

4.1. Formulation of the research task to evaluate the effectiveness of the proposed hybrid model for structured data storage

The effectiveness of the proposed hybrid model for storing structured data and the algorithm for dynamic data replication and verification is evaluated by comparing it with the two most common classes of existing approaches:

– *static replication models*, where the number and placement of replicas are fixed during network initialization and do not respond to topology changes:

– *decentralized DHT models for mobile P2P networks*, which rely exclusively on mathematical (spatial or ring) routing without centralized orchestration.

The data replication reliability metric used is the access success rate, defined as the percentage of requests for which data was found, passed integrity checks, and was successfully delivered to the initiator before the timeout expired.

4.2. Software Environment for Experiment Setup

To verify the proposed model and compare it with the most common existing approaches, a combined experimental setup was created. The setup includes three components.

1. *Mininet-WiFi network emulator*: allows creating virtual mobile nodes that support IEEE 802.11 protocols and integrate with SDN controllers [22, 23].

Network emulation parameters in Mininet-WiFi: 50 virtual nodes interacting in ad-hoc mode via the IEEE 802.11g wireless protocol; the radius of stable wireless coverage for each individual node is 250 meters; topology changes and spatial movement of devices are modeled using a stochastic random walk model, where the speed limits of the nodes range from 10 km/h to 100 km/h depending on the test scenario.

2. *OpenDaylight (ODL) SDN controller*: used to manage the control plane, implement routing algorithms, and orchestrate storage.

Telemetry data collection (remaining charge, motion vector, free memory) and transmission of replication commands to the DRL agent are implemented with a sampling rate of 500 ms.

3. *DRL framework (TensorFlow/PyTorch)*: implements agents for making decisions regarding adaptive replication and data migration.

The Deep Q-Network architecture [24, 25] is used as the neural network, with the following structure:

– an input layer (environment state vector) of dimension 4: current charge $B_i(t)$, memory quota C_i, *LET* time, and number of requests $AC_{i,k}$,

– two fully connected hidden layers with 128 and 64 neurons, respectively, using the ReLU activation function,

– a linear output layer for estimating Q-values of possible actions.

The network is trained using the Adam optimizer with a base learning rate of $\alpha_{LR} = 0.001$ and a reward discount factor of $\gamma_{DRL} = 0.95$.

4.3. Experimentl Methodology

The experiment is conducted in three stages.

1. Static replication testing: evaluation of basic integrity metrics for different numbers of replicas without node movement.
2. Dynamic replication testing: evaluation of time delays and routing overhead.
3. Evaluation of link stability for structured data under topology changes.

Topology dynamics were simulated by varying the average node movement speed. To ensure the reliability of the results, each experimental scenario (for each fixed node movement speed and each model under study) was repeated 30 times. The dispersion of indicators was assessed by calculating the standard deviation (*SD*). The results of the access success test when changing the node movement speed are presented in Table 3.

Table 3. **Results of static and dynamic replication testing**

Node movement speed scenario	Static replication, %	DHT, %	Proposed model, %
10 km/h (pedestrian traffic, occasional communication dropouts)	87.6, SD=1.8	92.6, SD=1.4	97.2, SD=0.6
20 km/h (bicycles/ scooters)	81.7, SD=2.4	86.3, SD=2.1	96.1, SD=0.8
60 km/h (city transport, cluster change)	66.4, SD=3.8	62.8, SD=4.2	91.4, SD=1.2
100 km/h (highway, high-intensity signal dropout)	46.9, SD=5.1	41.1, SD=5.8 (routing performance degradation)	83.9, SD=1.9

***Source*: compiled by the authors**

The lower *SD* value in the proposed model is explained by the adaptive preventive transfer of replicas within stable clusters, which reduces the stochastic impact of high-speed link failures.

Under low-mobility conditions (10 km/h), all studied systems demonstrate high reliability. However, as speed increases to 100 km/h, a significant drop in the performance of traditional architectures is observed. Static replication degrades to 46.9%, as nodes with data rapidly move out of radio range, and new copies are not created. Classic DHT protocols demonstrate even worse performance at 41.1 %. This is because high mobility causes logical pointer tables to become obsolete faster than algorithms can update them, resulting in requests getting lost among non-existent nodes.

The proposed hybrid model maintains a high availability rate of 83.9 % even at high speeds. This is achieved because the controller aggregates nodes into stable groups (e.g., a convoy of machines moving together) and performs forced migration of replicas to more central and power-supplied nodes (based on *WAC* and $S(i, j)$ metrics) even before the actual loss of signal.

Latency and routing overhead. The search time – the interval from the moment a request is formed to the moment the first valid bytes of information are received – is used as a metric for delays arising during data transmission. The results of evaluating time and energy metrics in structured data under topology changes are presented in Table 4.

Table 4. **Results of the stability assessment of time and energy metrics in structured data when changing the topology**

Metric	Static replication	DHT (Bamboo protocol)	Proposed model
Average search time (ms)	139 ms, SD=12.4 ms	207 ms, SD=18.7 ms	87 ms, SD=4.2 ms
Computa-tional algorithmic complexity	$O(1)$ (local) or O∞	$O(\log n)$ logical jumps	Virtual $O(1)$ (globally)
Response to a physical link failure	Time-consuming request reinitiation	Global DHT recalculation (from 500 ms)	Instantaneous software-defined routing (SDN)
Average communicat ion energy savings (Cost F)	Base level (0%)	up to 15%, SD=2.8%	up to 30%, SD=1.5% optimization

***Source*: compiled by the authors**

4.4. Analysis of the results

The analysis of delays demonstrates the shortcomings of DHT in MANET. According to research, a logical search in such systems requires $O(log\ n)$ logical hops between P2P peers. The underlying issue is that a single logical hop in the physical MANET topology may correspond to 3-5 physical hops through intermediate nodes. This generates cumulative delay, which averages 207 ms, and proportionally increases the probability of packet loss at each stage.

The proposed model is based on the SDS abstraction. At the application level, all data appears to be local (virtual $O(1)$). When an application accesses an array, the request is intercepted by the switch and redirected by the controller. Since the controller knows the global layout of the replicas, it calculates the mathematically optimal path and directs the packet directly to the nearest carrier, reducing the delay to 87 ms.

Regarding energy efficiency, adaptation based on stable neighbors reduces the average communication cost (*Cost F)* by 3 % compared to greedy algorithms. In addition, implementing a competitive aging mechanism instead of the classic LRU reduces the number of costly erase/write operations on the flash storage by 50 %, which significantly extends the service life of mobile device hardware and preserves battery life.

5. DISCUSSION OF RESULTS

The presented experimental results demonstrate that managing structured data sets in highly dynamic stochastic environments cannot be effectively based solely on decentralized mathematical constructs (P2P DHT) or static cloud paradigms. Achieving stability requires a deep hybrid approach, which is proposed within the scope of this work.

The scientific novelty of this work lies in the development of a model for preserving structured data using modern SDN network paradigms and heuristic adaptive replication with competitive aging mechanisms, which has resolved the established contradiction for mobile specialized networks and improved the level of availability and integrity of information in distributed programmable mobile systems under conditions of dynamic network topology and limited resources of mobile nodes.

The proposed approach to storing structured data in distributed programmable mobile systems has several advantages. First, thanks to the application of the software-defined storage methodology in the MANET context, the strict hardware limitations of mobile devices (small disks, various interfaces) do not constrain the system; memory is dynamically allocated by the controller as a single space.

Second, a mathematical criterion is proposed for calculating the “suitability” of candidate nodes for hosting replicas. Unlike classical approaches, this criterion integrates not only access frequency but also applies an energy penalty (weighted available charge WAC) as well as a predictive stochastic metric of the stability of the neighbor graph $S(i, j)$. This ensures the migration of data sets to the most energy- and topologically-stable network nodes even before the onset of the physical radio link failure phase.

Third, the use of a dual-transaction interface (“strong” and “weak”) formalized the concept of controlled or limited inconsistency during outages. This approach facilitated the uninterrupted operation of applications even in isolated segments with subsequent automatic synchronization, without compromising the global semantic integrity of the system’s data in the long term.

The theoretical developments and empirical results obtained have great potential for practical implementation in a range of social and industrial projects.

1. Mitigating the consequences of large-scale disasters. In regions where the basic fiber-optic or cellular infrastructure has been physically destroyed (e.g., by earthquakes or military operations), units rely on autonomous MANET networks. The proposed hybrid model ensures that critically important structured information (e.g., topographic maps, coordinates of friendly units, medical records of the wounded) does not disappear without a trace along with a disabled or remote communication node. Thanks to the WAC algorithm, data will be replicated in advance and transparently to users on the most reliable nearby devices.

2. Intelligent Transportation Systems and Smart Cities. In Vehicle-to-Everything (VANET) networks, vehicles change the network topology extremely quickly and chaotically. The model will allow streams of vehicles to instantly exchange critical data sets (for example, regarding emergency obstacles ahead), relying on stable clusters (columns of cars with the same direction of travel) detected by the SDN controller, minimizing unnecessary load on base stations and ensuring ultra-low access latency (87 ms), which is critical for autonomous braking systems.

3. Distributed offices and remote manufacturing. Software-Defined Factories face the challenge of integrating devices from different

administrative domains into a single process. SDS architecture allows such remote branches to use existing standard equipment to create highly fault-tolerant storage clusters without resorting to expensive solutions.

CONCLUSIONS

An analysis of the nature of network vulnerabilities – including frequent wireless connection drops, the chronic shortage of computational and power resources in portable devices, and the inherent trade-offs between semantic consistency and information availability – has demonstrated that existing decentralized models, in particular, traditional distributed hash tables (DHTs such as Chord or Bamboo), static replication algorithms, and isolated edge file systems, are unable to effectively cope with the phenomenon of mass node entry, exit, and spatial movement. In critical situations, they generate excessive service traffic, draining battery resources, overloading storage devices, and allowing for the degradation of logical data integrity.

To address this problem, this paper proposes a model for storing structured data in distributed programmable mobile systems. The model is based on the synthesis of the flexibility of software-defined networks (SDN) with the abstractions of logical spaces in software-defined storage (SDS).

The developed hybrid model enables the implementation of intelligent data fragmentation. Prioritizing structural elements (metadata) over content ensures the logical integrity of the system even in the event of significant degradation of the physical network layer, which is critical for emergency response systems.

To optimize the proposed model, an adaptive dynamic replication algorithm has been developed that operates based on the calculation of complex stability metrics of neighboring nodes, weighted by the energy penalty of access patterns and mechanisms for competitive replacement of old files.

The results of experimental studies demonstrate the superiority of the proposed hybrid model. The model is capable of maintaining a request delivery success rate of over 84.5% even under conditions of massive topology shifts at a speed of 100 km/h, which is twice the capability of classical P2P DHT protocols with their 40.5 %. In addition, the hybrid model significantly reduces the access time to the required structured data from 210 ms to 85 ms by eliminating the need for multiple logical P2P hops and enabling direct packet routing by a centralized logical controller exclusively over the shortest and most stable physical links.

The use of a two-level hybrid consistency model supporting “weak” and “strong” transactions has made it possible to increase the level of continuous data availability during full or partial topology splits, while maintaining guaranteed delayed semantic integrity of the system as a whole.

These findings open up extensive opportunities for implementation in autonomous transportation and industrial mobile systems.

USE OF ARTIFICIAL INTELLIGENCE

While preparing this article, the authors used Google Gemini to verify the logical consistency of the text in order to improve its readability and ensure that bibliographic descriptions comply with international standards. The authors bear sole responsibility for the content of this publication.

REFERENCES

1. Rathod, V. U. & Gumaste, S. V. “Role of Deep Learning in Mobile Ad-hoc Networks”. *International Journal on Recent and Innovation Trends in Computing and Communication*. 2022; 10 (2s): 237–246. DOI: https://doi.org/10.17762/ijritcc.v10i2s.5938.

2. Kampitaki, D. G. & Economides, A. A. “Selfishness in Mobile Ad-Hoc Networks: A Literature Review on Detection Techniques and Prevention Mechanisms”. *IEEE Access*. 2023; 11: 86895–86909, https://www.scopus.com/pages/publications/85168274903.
DOI: https://doi.org/10.1109/ACCESS.2023.3305262.

4. Aldossary, D., Aldahasi, E., Balharith, T. & Helmy, T. “A systematic literature review on load-balancing techniques in fog computing: Architectures, strategies, and emerging trends”. *Computers*. 2025; 14 (6): 217, https://www.scopus.com/pages/publications/105008894535.
DOI: https://doi.org/10.3390/computers14060217.

4. Tamuka, N., Mathonsi, T. E., Olwal, T. O., Maswikaneng, S., Muchenje, T., & Tshilongamulenzhe, T. M. “Intrusion detection in fog computing: A systematic review of security advances and challenges”. *Computers*. 2026; 15 (3): 169, https://www.scopus.com/pages/publications/105033860038. DOI: https://doi.org/10.3390/computers15030169.

5. Vinueza-Naranjo, P. G., Chicaiza, J. & Rumipamba-Zambrano, R. “Fog computing technology research: A retrospective overview and bibliometric analysis”. *ACM Computing Surveys*. 2024; 57 (4): 1–32, https://www.scopus.com/pages/publications/85214837862. DOI: https://doi.org/10.1145/3702313.

6. Rajesh, R. & Patgiri, R. “Survey based on edge structured file systems in edge computing”. *Tehnički glasnik*. 2025; 19 (2): 305–312, https://www.scopus.com/pages/publications/105002421678. DOI: https://doi.org/10.31803/tg-20240111081254.

7. Carvalho, G. et al. “A survey of holistic approaches for distributed database systems: From conceptual model to deployment”. *IEEE Access*. 2025; 13: 120831–120834, https://www.scopus.com/pages/publications/105010533025.
DOI: https://doi.org/10.1109/ACCESS.2025.3587670.

8. Alubady, R. “A review of modern caching strategies in named data networks: overview, classification, and research directions”. *Telecommunication Systems*. 2023; 84 (4): 1–46, https://www.scopus.com/pages/publications/85170214796. DOI: https://doi.org/ 10.1007/s11235-023-01015-3.

9. Elouataoui, W. & Gahi, Y. “Empirical evaluation of big data stacks: Performance and design analysis of hadoop, modern, and cloud architectures”. *Big Data and Cognitive Computing*. 2026; 10 (1): 7, https://www.scopus.com/pages/publications/105028509251. DOI: https://doi.org/10.3390/bdcc10010007.

10. Vera, H., Guo, R., Holanda, M. & Mariano, A. M. “Data modeling and NoSQL databases – A systematic mapping review”. *ACM Computing Surveys*. 2021; 54 (6): 1–26. DOI: https://doi.org/10.1145/3457608.

11. Brahmia, Z., Grandi, F. & Oliboni, B. “A literature review on schema evolution in databases”. *Computing Open*. 2024; 2: 2430001–2430029. DOI: https://doi.org/10.1142/S2972370124300012.

12. Bayazitov, D., Kozhakhmet, K., Omirali ,A. & Zhumaliyeva, R. “Leveraging Amazon Web Services for cloud storage and AI algorithm integration: A comprehensive analysis”. *Applied Mathematics & Information Sciences*. 2024; 18 (6): 1235–1246, https://www.scopus.com/pages/publications/85204125283. DOI: https://doi.org/: 10.18576/amis/180606.

13. Boughdiri, M., Abdellatif, T. & Guégan, C. “A systematic literature review on blockchain storage scalability”. *IEEE Access*. 2025; 13: 102194–102219. DOI: https://doi.org/10.1109/ACCESS.2025.3578451.

14. Ahmad, S. & Mir, A. H. “Scalability, consistency, reliability, and security in SDN controllers: A survey of diverse SDN controllers”. *Journal of Network and Systems Management*. 2021; 29 (1), https://www.scopus.com/pages/publications/85095699320. DOI: https://doi.org/10.1007/s10922-020-09575-4.

15. Huang, Y.-X. & Chou, J. “A survey of NFV network acceleration from the ETSI perspective”. *Electronics*. 2022; 11 (9), https://www.scopus.com/pages/publications/85129166172.
DOI: https://doi.org/10.3390/electronics11091457.

16. Muñoz-Escoí, F. D., Juan-Marín, R., García-Escrivá, J.-R., González de Mendívil J. R. & Bernabéu-Aubán J. M. “CAP theorem: Revision of its related consistency models”. *The Computer Journal*. 2019; 62 (6): 943-960, https://www.scopus.com/pages/publications/85068524207.
DOI: https://doi.org/10.1093/comjnl/bxy142.

17. Nesterenko, S. A., Tishinm, P. M., Shtilmanm, P. R., Martynyuk, O. N. & Mileiko, I. G. “Fuzzy models of wireless component sensor networks”. *Herald of Advanced Information Technology*. 2025; 8 (2): 209–220. DOI: https://doi.org/10.15276/hait.08.2025.13.

18. Abid, A., Lee, J., Le, Gall F. & Song, J. “Toward mapping an NGSI-LD context model on RDF graph approaches: A comparison study”. *Sensors*. 2022; 22 (13): 4798, https://www.scopus.com/pages/publications/85132777682. DOI: https://doi.org/10.3390/s22134798.

19. Drobnič, F., Starc, G., Jurak, G., Kos, A. & Pustišek, M. “Automatic generation of NGSI-LD data models from RDF ontologies: Developmental studies of children and adolescents use case”. *Applied Sciences*. 2026; 16 (2): 992, https://www.scopus.com/pages/publications/105028794438. DOI: https://doi.org/ 10.3390/app16020992.

20. Rahmani, A. M., et al. “An adaptive and multi-path greedy perimeter stateless routing protocol in flying ad hoc networks”. *Vehicular Communications*. 2024; 50: 100838, https://www.scopus.com/pages/publications/85203514323.
DOI: https://doi.org/10.1016/j.vehcom.2024.100838.

21. Yun, S. et al. “i-CU: Intelligent cache replacement and content update for data freshness in cloud-edge networks”. *IEEE Transactions on Mobile Computing*. 2025: 24 (12): 12742−12755, https://www.scopus.com/pages/publications/105012447874. DOI: https://doi.org/10.1109/TMC.2025.3589609.

22. Surkov, S. S., Martynyuk, O. M., Drozd, O. V. & Drozd, M. O. “A model and method for enhancing the efficiency of processing operation queues at maximum server equipment load.” *Applied Aspects of Information Technology*. 2024; 7 (2): 125–134. DOI: https://doi.org/10.15276/aait.07.2024.9.

23. Zahid, S., et al. “Fault tolerant DHT-based routing in MANET”. *Sensors*. 2022; 22 (11): 4280, https://www.scopus.com/pages/publications/85131168101. DOI: https://doi.org/10.3390/s22114280.

24. Marouane, C. & Saad, B. “Safe Navigation based on deep Q-Network algorithm using an improved control architecture”. *2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC)*, Setif, Algeria. 2024. p. 1–6, https://www.scopus.com/pages/publications/85199005185. DOI: https://doi.org/10.1109/ICEEAC61226.2024.10576248.

25. Kiyaei, M. & Kiaee, F. “Optimal ATM cash replenishment planning in a smart city using deep Q-network”. *26th International Computer Conference, Computer Society of Iran (CSICC)*. Tehran: Iran. 2021. p. 1–5, https://www.scopus.com/pages/publications/85106201799.
DOI: https://doi.org/10.1109/CSICC52343.2021.9420561.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 26.03.2026
Received after revision 29.05.2026
Accepted 10.06.2026

DOI: https://doi.org/10.15276/aait.09.2026.23
UDC 004.75

Модель збереження структурованих даних у гетерогенних розподілених програмованих мобільних системах

Фомін Олександр Олексійович[1)]
ORCID: https://orcid.org/0000-0002-8816-0652; fomin@op.edu.ua. Scopus Author ID: 57103429400
Чмелевський Андрій Миколайович[1)]
ORCID: https://orcid.org/0009-0008-6450-6875; stech@stud.op.edu.ua
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Стрімкий розвиток мобільних мереж спеціалізованого призначення та периферійних обчислень вимагає створення стійких архітектур зберігання даних, здатних ефективно функціонувати в умовах стохастичної зміни топології. Існуючі децентралізовані підходи демонструють високі накладні витрати та деградацію цілісності при критичних розривах зв'язку, що робить адаптивну організацію інформаційного обміну **актуальним** та критично важливим завданням. Робота присвячена вирішенню суперечності між необхідністю забезпечення цілісності та високої доступності інформації з одного боку, та жорсткими обмеженнями ресурсів мобільних вузлів (енергоспоживання, обсягу пам’яті та пропускної здатності каналів зв’язку) з іншого боку в умовах постійної зміни топології та нестабільності каналів зв’язку розподіленої системи. **Метою статті** є забезпечення високої доступності та цілісності інформації у розподілених програмованих мобільних системах в умовах динамічної топології мережі та обмежених ресурсів мобільних вузлів. Для досягнення цієї мети поставлено такі **завдання**: запропонувати гібридну модель збереження структурованих даних на основі сучасних семантичних стандартів; розробити алгоритм адаптивної реплікації із прогнозуванням траєкторії руху пристроїв; експериментально оцінити ефективність розроблених рішень. Використано такі **методи**: теорія графів для моделювання структури мережі, апарат програмно-визначених мереж та програмно-визначеного зберігання для оркестрації ресурсів, а також методи математичного моделювання для розрахунку очікуваного часу життя бездротових з'єднань. **Наукова новизна** роботи полягає в удосконаленні моделі збереження структурованих даних із застосуванням парадигми програмно-визначених мереж та евристичної адаптивної реплікації з механізмами конкурентного старіння, що дозволило вирішити встановлене протиріччя. **Практична значимість** отриманих результатів полягає у можливості їх безпосереднього впровадження в інтелектуальних транспортних системах, смарт-містах та мережах екстреного реагування під час надзвичайних ситуацій. Найбільш важливі **результати** дослідження полягають у утриманні запропонованою моделлю коефіцієнта успішності доставки запитів на рівні вісімдесят трьох цілих дев’яти десятих відсотка при швидкостях до ста

кілометрів на годину, скорочуючи середній час пошуку з двісті семи до вісімдесят семи мілісекунд. **Висновки:** створення гібридної моделі на основі інтелектуальної оркестрації та дворівневої консистентності дозволяє забезпечити режим контрольованої неузгодженості, забезпечуючи безперебійну роботу прикладних додатків у розірваних сегментах із гарантованим відновленням глобальної цілісності даних у довгостроковій перспективі.

Ключові слова: моделювання; мобільні спеціалізовані мережі; гетерогенні системи; структуровані дані; реплікація даних; динамічна топологія; периферійні обчислення, узгодженість даних

ABOUT THE AUTHORS

Oleksandr O. Fomin - Doctor of Engineering Sciences, Professor, Department of Computerized Systems and Software Technologies. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-8816-0652; fomin@op.edu.ua. Scopus Author ID: 57103429400
***Research field*:** mathematical modeling, information systems, intelligent control systems

Фомін Олександр Олексійович - доктор технічних наук, професор кафедри Комп'ютеризованих систем та програмних технологій. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Andriy M. Chmelevskyi - PhD student, Department of Computer Systems. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0009-0008-6450-6875; stech@stud.op.edu.ua.
Research field: computer engineering, control of mobile objects

Чмелевський Андрій Миколайович - аспірант, кафедра Комп'ютерних систем. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна